AD

CONTRACT REPORT ARBRL-CR-00483

ANALYSIS OF MODULATED OSCILLATOR DATA

Prepared by

Falcon Research and Development Company
696 Fairmount Avenue
Baltimore, MD 21204

July 1982

US ARMY ARMAMENT RESEARCH AND DEVELOPMENT COMMAND
BALLISTIC RESEARCH LABORATORY
ABERDEEN PROVING GROUND, MARYLAND

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER Contract Report ARBRL-CR-00483	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) ANALYSIS OF MODULATED OSCILLATOR DATA		5. TYPE OF REPORT & PERIOD COVERED Contract Report
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) R. S. Bennett N. E. Aldridge M. E. Delkhoon		8. CONTRACT OR GRANT NUMBER(s) N00123-77-C-0549
9. PERFORMING ORGANIZATION NAME AND ADDRESS Falcon Research and Development Company 696 Fairmount Avenue Baltimore, MD 21204		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 1L162618AH80
11. CONTROLLING OFFICE NAME AND ADDRESS U.S. Army Armament Research & Development Command U.S. Army Ballistic Research Laboratory (DRDAR-BL) Aberdeen Proving Ground, MD 21005		12. REPORT DATE July 1982
		13. NUMBER OF PAGES 148
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved For Public Release; Distribution Unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Projectile-Gun Dynamics
Projectile Launch
Data Analysis

20. ABSTRACT (Continue on reverse side if necessary and identify by block number) jmk

A numerical data analysis technique which permits the evaluation of time series recorded by a radio frequency oscillator measurement system mounted at the gun muzzle is described. The report briefly outlines the measurement method and describes in detail the requirements on the data analysis, the analysis methodology, the developed computer program and its operation.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

ANALYSIS OF MODULATED OSCILLATOR DATA

1. INTRODUCTION

The Interior Ballistics Division of the Ballistic Research Laboratory has conducted a firing experiment with a fully instrumented cannon in order to establish and maintain a technology base for the study of projectile-gun dynamics[1]. Of particular importance is the investigation of problems associated with the erratic flight behavior of certain projectiles.

The investigation of projectile in-bore and launch dynamics involved a number of firings of an M-68 105mm tank cannon during the fall and winter of 1977. The instrumentation of this cannon included an inductance coil mounted at the muzzle face and forming part of a Hartly radio frequency oscillator. This coil and oscillator forms the sensor portion of a measurement system which is intended to provide an accurate muzzle exit time, monitor projectile integrity, determine the projectile velocity, and trigger auxiliary equipment. Determining the precise time of projectile exit and the duration of the passage of the projectile through the coil are the primary functions of this instrument.

The complete muzzle velocity measurement system consists of three basic units: the sensor, the signal processor, and the data processor. The sensor provides a pulse with amplitude proportional to the geometric shape of the projectile. The signal processor amplifies the detected pulse, filters and samples it, and prepares it for data processing. The data processor continues with the extraction of time and speed information from the raw data.

The sensor is a single loop inductance coil which has been etched on a single clad printed circuit board and is rigidly attached to the gun tube muzzle. The configuration shown in Figure 1.1, allows the sensor to be mounted without drilling holes into the gun tube. The sensor coil is part of the tuned circuit of the Hartly oscillator and therefore directly affects both the amplitude and frequency of oscillation.

An expanded view of the sensor coil arrangement is shown in Figure 1.2. Its components are the guard ring, sensor coil, and teflon seal. The inductance coil is insulated and separated from the muzzle face by the thickness of the board

---

[1] *R. K. Loder, J. O. Pilcher, "Nondestructive Test Method to Establish the Performance of Projectile-Gun Systems," Proceedings of the 26-th Defense Conference on Nondestructive Testing, 15 to 17 November 1977, Seattle, Washington; Published by U.S. Army Materials and Mechanics Research Center, Watertown, MA (pages 272-303).*

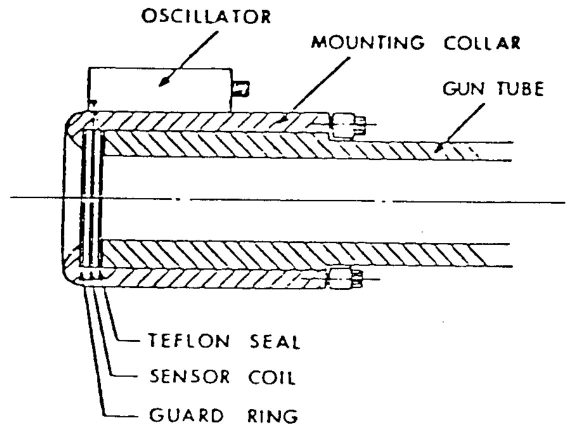


Figure 1.1 Cross-Sectional View of Sensor Coil Assembly Rigidly Attached to M-68 Gun Tube Muzzle

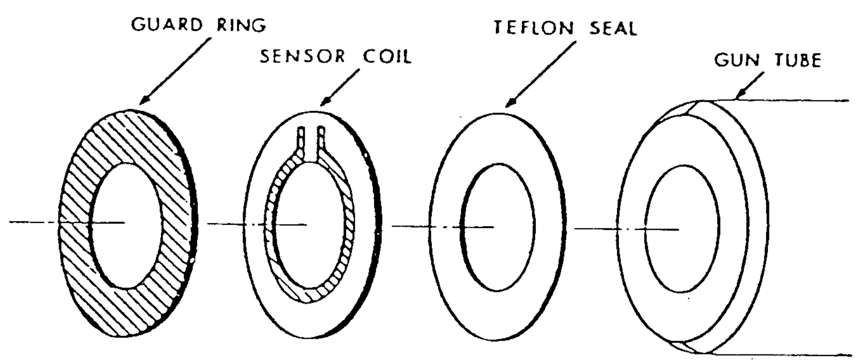


Figure 1.2 Sensor Coil Expanded View

from which it is made. A second ring consisting of an unetched single clad circuit board forms a shield (guard ring) and is mounted in front of the sensor coil. Both pieces of circuit board are affixed with resin to the front face of the collar and securely held to the gun tube muzzle by the mounting collar. To prevent high pressure gas from leaking between the muzzle face and the sensor coil ring, a teflon seal had to be inserted. The rest of the oscillator circuit is housed in a small box mounted on the collar assembly.

Electronically, the oscillator is a conventional Hartly circuit. The utilization of this oscillator for monitoring the projectile exit from the gun muzzle is based on the recording and evaluation of changes in the electric current of the oscillator which are caused by the interaction of the electromagnetic field of the inductance coil with the transient in-bore environment consisting of the moving projectile and the gas flow which surrounds it.

Ideally, the oscillator coil would have a constant inductance and distributed capacitance prior to the arrival of the projectile, and then would show sharp changes due to the magnetic and dielectric properties of the projectile material. Thus the desired time information could be easily and accurately extracted by simply comparing the sampled values on the tape with a fixed threshold. More detailed analysis would yield a profile of the projectile cross section as a time function, or the relative positions of salient features of the projectile.

Since the amplitude of the oscillator current is modulated as a function of the contour of the metallic shell of the projectile, any discontinuity in the projectile contour can be chosen as a geometric reference point. By correlating the passage time of the reference points with their spatial displacements, one can determine the muzzle exit velocity of the projectile.

In practice, the coil and oscillator sense the approach of the projectile before it arrives at the muzzle, thus "softening" the change in oscillator output at the time of exit. The inductance and capacitance of the coil are affected by a number of phenomena other than the passage of the projectile[2]. Gases within the coil prior to and following the projectile exit have an effect on the coil electrical properties. Furthermore, the dimensions of the coil and its proximity to the muzzle structure

---

[2] *R. K. Loder, J. Q. Schmidt, "Radio Frequency Oscillator Technique for Monitoring Velocity and Structural Integrity of Projectiles from Their Exit from the Muzzle," ARBRL-MR-03100, USA ARRADCOM, Ballistic Research Laboratory, Aberdeen Proving Ground, MD, April 1981.* (AD A100725)

vary slightly during firing due to vibration of the tube and coil mounting, and thus the vibration is translated into a time-varying inductance change. As it is always the case with electronic devices, the circuitry associated with the coil and oscillator adds a random fluctuation, or instrumentation noise, component to the signal. Generally, the vibration produces a quasiperiodic component in the recorded signal, while the instrumentation noise is nearly random. In actuality, the vibration period is not constant, and the noise does not necessarily have a Gaussian amplitude distribution or flat power spectrum. Thus very simple filtering and smoothing are not sufficient to remove these corrupting components.

The instrumentation noise, gas, and vibration effects on the signal tend to even further obscure the points where the projectile arrives at and departs from the coil. As an additional problem, when severe saturation of the electronics occurs as a result of the signal exceeding the dynamic range of the circuitry, the relatively slow decaying transients produced distort the subsequent waveform to the extent that the data are useless. In some cases, exceeding the dynamic range for very brief periods causes only minor clipping of the waveform, but even in these cases some data are lost.

While the gas and vibration effects are a problem when determination of exit time and duration are to be made, these same effects permit the recovery of some secondary data from the oscillator waveforms. The secondary data include measurement of radial and/or longitudinal gun tube vibration, and the location of the shock front.

The most important of the coil loading phenomena, for the purpose here, is the electromagnetic field induced in the projectile surface by the electromagnetic field of the oscillator. The physical process which underlies this major loading source is described as follows: The varying electromagnetic field radiated from the active inductance coil produces localized eddy currents in the metallic surface of the projectile when it moves through the coil. The eddy currents, in turn, generate an electromagnetic field which couples back into the active inductance coil. The closer the metallic shell of the projectile is to the inductance coil, the stronger is the induced secondary field and the effect of its recoupling to the inductance coil. The variation in the oscillator impedance causes an increase or decrease of the operating frequency of the oscillator which can be recorded. This frequency modulation was the basis for a measurement device developed by Mr. G. Schultze at ISL[3]. However, with the proper

---

[3] *G. Schultze, "Telemetrische V_o-Messungen"; Colloquium on Ballistic Measurement Techniques; held at the Institut Franco-Allemand De Recherches De Saint-Louis, 6 and 7 June 1973; ISL Report 14/73.*

selection of oscillator circuit parameters, the change in impedance can be kept such that the frequency shift is minimized and the amplitude modulation is maximized for obtaining the signature of the projectile exit. This technique of amplitude modulation was selected by Mr. J. Q. Schmidt[4] for the M-68 instrumentation, and is the technique for which data are evaluated by the program described in this report.

Falcon Research and Development Company has generated a computer code, designated as "MOdulated OScillator Evaluation", (MOOSE), which uses the 2.5 microsecond interval samples from the detected oscillator signal as its input, and performs the following operations:

1. Classification of the recorded time series according to the quantity of data lost due to saturation:

   a. No data loss.
   b. Slight loss due to clipping during pass-through.
   c. Major section of the history lost following saturation during pass-through.

2. Determination of the beginning and end of pass-through.

3. Division of the time series into three regimes:

   a. Prior to projectile exit; Regime I.
   b. During projectile pass-through; Regime II.
   c. Following pass-through; Regime III.

4. Data smoothing by local mean or local median.

5. Determination of the power spectra for Regimes I and III.

6. Description of the signal in Regimes I and III as a sum of three major components:

   a. An aperiodic function which is modeled as a 4^{th} degree polynomial.
   b. A periodic component which is modeled as the sum of ten sinusoidal functions, harmonically related but not necessarily by consecutive harmonics.
   c. Random noise.
   d. Identification of shock-like waveforms.

---

[4] *J. Q. Schmidt, "A Radio Frequency Oscillator Technique for Measuring Projectile Muzzle Velocity," USARRADCOM, Ballistic Research Laboratory, Aberdeen Proving Ground, MD; Technical Report ARBRL-TR-02158, April 1979. (AD B038926L)*

Note: As an option, the first regime may be divided into two parts, with the division point being that point where the periodic component changes its fundamental frequency. The two parts of the first regime are then analyzed in the manner described for the complete regime.

7. Determination of the polynomial coefficients by an orthogonalized polynomial technique.

8. Determination of the ten sinusoidal frequencies by choosing the frequencies for which the power spectrum shows the greatest energy.

9. Checking the random component for nonperiodicity and "whiteness" by power spectrum analysis, and for the presence of a Gaussian amplitude distribution by a Chi Square (χ^2) test.

10. Determination, for Regime II, of the following:

    a. Extrema.
    b. Power Spectra.

The program was prepared in FORTRAN and is intended for use at the BRL computer facility at Aberdeen Proving Ground. In the preparation of the program, extensive use was made of the IMSL subroutines in order to minimize coding time. The mathematical analysis performed by MOOSE, and the computer code itself, are fully described in the following sections.

2. ELEMENTS OF ANALYSIS

2.1 Data Description.

Measurements were taken by the induction coil and oscillator for a total of five milliseconds, beginning approximately at the time of ignition. The time interval between data points is 2.5 microseconds. The data are in units proportional to the demodulated signal voltage and have values ranging between approximately 1800 and -2300 units, although higher values did occur. When the signal exceeded these limits for more than 25 microseconds, saturation of the instrumentation led to a loss of subsequent data.

During the time period in which the induction coil operates, three stages of data are represented. The first is just after ignition, while the projectile is still completely in the tube. During the second stage, the projectile is exiting the tube, and the third is after the projectile has completely emerged from the tube. The signal shape for each of these stages is distinct, although, for the various reasons discussed in the introduction the divisions between them are not always well defined. For the purposes of analysis, each of the physical stages of the firing corresponds to a regime in the data set.

The first regime begins with a noisy, relatively constant signal at or below the zero level. In some cases a strong periodic signal is introduced during regime one, generally accompanied by a shock signal.

The second regime, using the proof slug data for illustration purposes, is marked by a sharp rise in the level of the signal, well in excess of the average noise level, as the projectile enters the induction coil. The signal peaks and decreases rapidly by approximately one third. This spike is the most prominant feature in the data set, although it does not necessarily represent the maximum signal value. Following the signal decrease, there is a period of constant value followed by a second, but smaller spike. The signal immediately plummets as the projectile leaves the coil, perhaps evincing one or two spikes during the descent, depending on the cross section of the projectile. The signal, after reaching its minimum, moves directly into the third regime. This minimum is generally not far from the baseline signal level in regime one. Conventional types of projectiles show a different form for the second regime. The noisy interval between the two peaks is reduced to a point or two and its level is very near that of the two peaks. In addition, no spikes are exhibited as the signal attains its minimum at the end of the regime.

Finally, after the projectile has completely emerged from the gun barrel, the induction coil measures the signals still being produced by gases and vibrations. This is in Regime III of the data set. There are three basic components to this interval. The first is the baseline which rises from the minimum at the last point in the second regime and eventually either decreases or levels off. Superimposed on this baseline is a well-defined periodic signal. Instrumentation noise occurs throughout Regime III, possibly punctuated by shocks.

Representative data sets are plotted in Figure 2.1, showing typical characteristics of Regimes I, II, and III, and the differences in the signals of two types of projectiles.

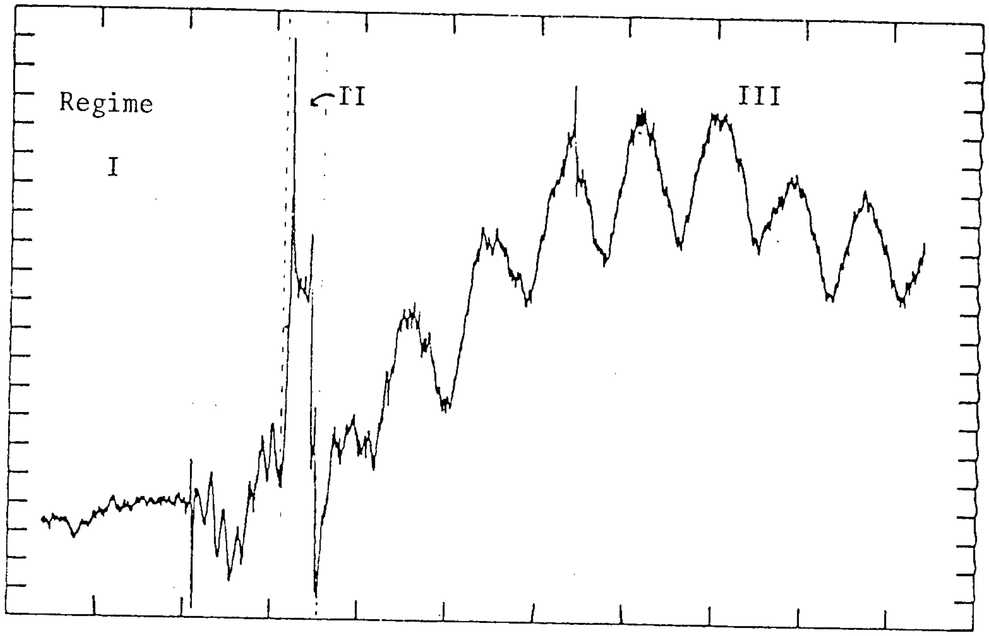
2.2 Requirements of Analysis.

As discussed in the previous section, there are several distinct signal components present in the induction coil data, each representing some physical process in the gun firing. It is assumed that these components add linearly, although in some cases, such as vibration of the coil windings, some multiplicative combinations may occur. The requirements for separating the signal components and the implications of these requirements are discussed in the following paragraphs.

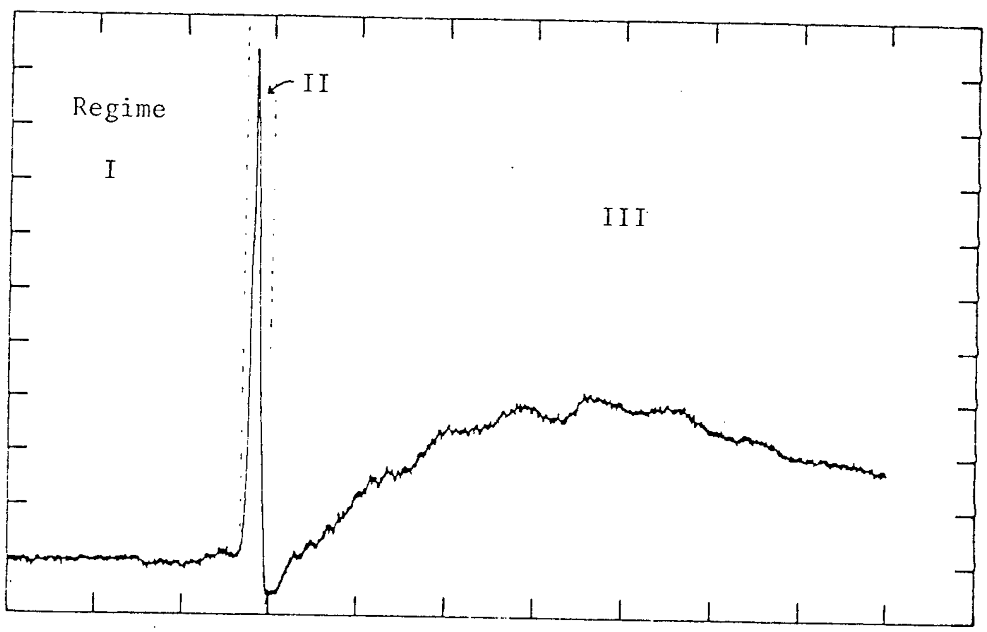
The first requirement is that the computer program automatically identify any data sets in which the instrumentation became overloaded or saturated. A decision is then necessary as to the extent of the analysis which is possible for that data set, and the amount of data which is unaffected by the saturation.

After the identification of incomplete or "damaged" data sets, the next function of the analysis is to determine the duration of Regime II, which is the time taken by the projectile to pass the induction coil. Identification of the epochs defining the end points of Regime II also, of course, determines the end of Regime I and beginning of Regime III. Because knowledge of the epoch locations is critical to the calculation of muzzle velocity, an estimate should be made of the error expected in determining their locations.

The next priorities are the analyses of the first and third regime. Regime I may be divided into two subregimes characterized by a pronounced shift in the fundamental frequency of the periodic component, but the same general process is used on each data interval. This analysis separates the four components of the signal. The first component is the trend, or baseline of the interval. The second is the periodic signal. Last are the noise and the occasional shocks. Equations are to be found to describe the baseline and the periodic component. The noise is described in terms of its spectrum and distribution of the amplitudes, rather than by an equation. Points that may represent shocks are indicated.



Calibration Shot (proof slug)



Conventional Round (M392)

Figure 2.1 Representative Data Sets

Nonrandom trends or periodicities in the "noise" may reveal inadequacies in the derivation of the baseline and periodic function of a regime.

Finally, there may be two different portions to Regime I. The transit of the shock front of the compressed air in front of the projectile may manifest itself as a shock in the signal. When this occurs, the signal waveform succeeding the shock shows more baseline variation and periodicities than the anteceding one. To adequately model Regime I, it may be advantageous to apportion it into two parts, with each analyzed separately. The requirement for accuracy in detecting this epoch is less critical since each subregime can be adequately modeled and analyzed even when the chosen division point is slightly in error.

Table 2-1 summarizes all of the requirements of the analysis for the data.

TABLE 2-1 REQUIREMENTS FOR THE ANALYSIS OF THE OSCILLATOR SIGNALS.

1.	Classification into good or "damaged" data sets.
2.	Length of time in Regime II.
3.	Periodicities of Regimes I and III.
4.	Description of noise of Regimes I and III.
5.	Baseline of signal of Regimes I and III.
6.	Relative location of shocks.
7.	Accuracy of epoch identification.
8.	Location of preliminary shock in Regime I.
9.	Power spectrum and extrema of Regime II.

2.3 APPROACH.

In this section, the major assumptions underlying the approach to the four major phases of the analysis of the oscillator signals will be discussed. These four phases include identification of incomplete or damaged data records, location of the transition points between the three data regimes, analysis of the data in each regime, and analysis of the residual noise.

Identification of damaged data records is made by the algorithm shown in Figure 2.2. This algorithm essentially looks for indications that the recording electronics became saturated during Regime II, that is, during the passage of the projectile through the induction coil. It is assumed that the saturation signal will have the characteristics which this routine detects, in particular that successive data points will have the same amplitude during the saturation portion of the record. A further assumption made for this case is that short-term saturation, less than 10 points or 25 microseconds, or "clipping" will have relatively little effect on the signal following the return of the recorded values into the linear range of the instrumentation. Furthermore, it is assumed that heavy saturation, as detected by this algorithm, will cause transients in the instrumentation output which render the remaining portion of the record useless, and the MOOSE program automatically analyzes only the first regime.

The second phase of the analysis is the identification of the transition points between Regimes I and II and between Regimes II and III. These transition points, or "epochs", are produced when the projectile enters and leaves the induction coil, and thus represent transitions between three physically distinct portions of the data. Detection of the epochs is the essential operation in determining the projectile exit velocity, and for providing key timing information for analyzing the data from other instrumentation. It must be assumed that the epochs can be determined with the accuracy needed for these purposes by either operator-assisted interactive means or by fully automatic techniques, and furthermore it is assumed that no data other than that provided on the oscillator coil record is needed for the determination.

Research in signal analysis has been concerned with epoch detection of this type for many years. In general, the approach has been to fit the data with a function, or sum of functions, and then for each point in the record extrapolate the data both forward and backward. An epoch caused when a new signal is added to the existing one will be indicated by a large discrepancy between the values obtained by forward and backward extrapolation. This approach works well when the signals are sums of exponentials and the noise is Gaussian. This technique was attempted with the gun data but was found

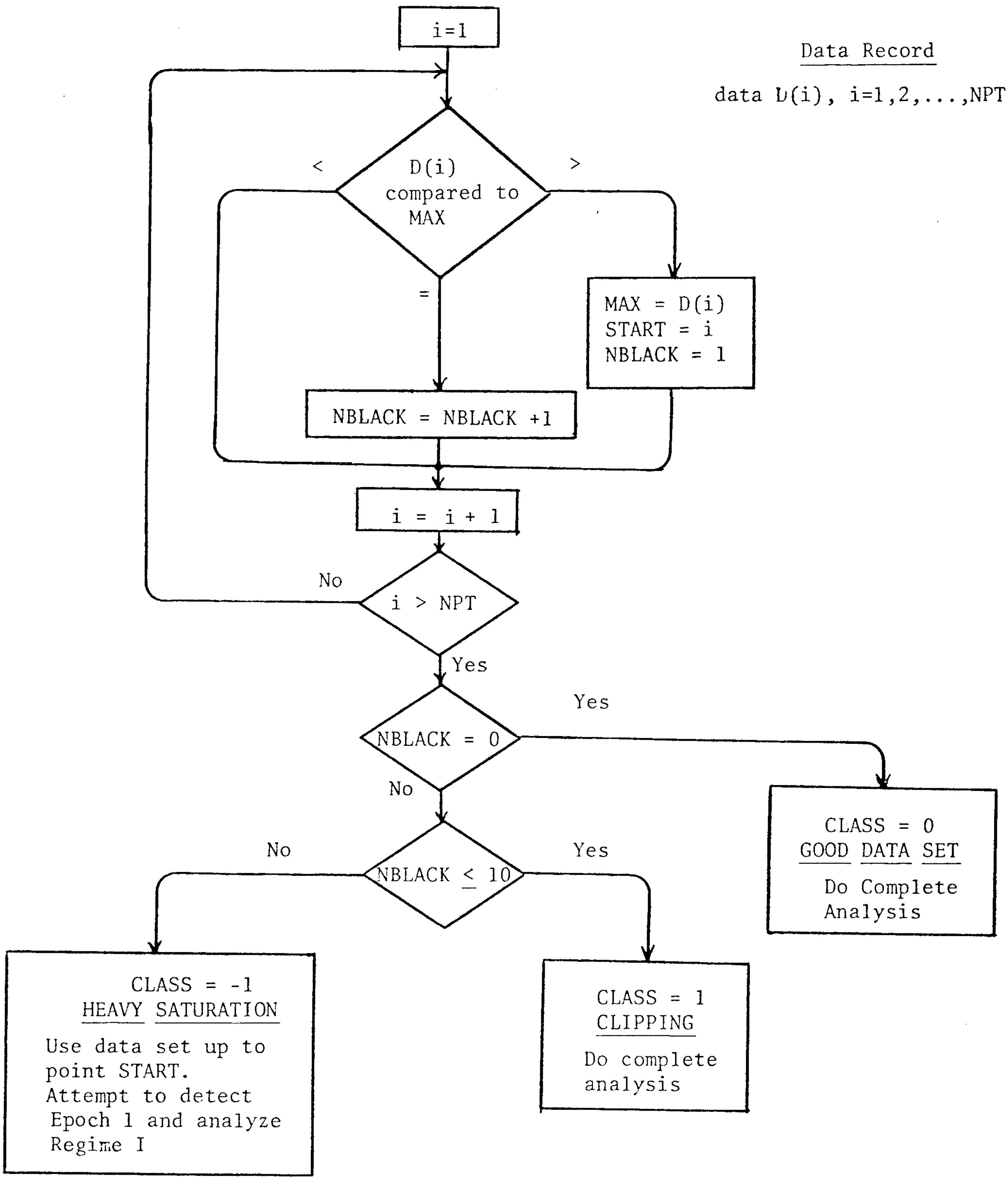


Figure 2.2 Data Classification Algorithm

to give erratic results. A technique based on measurement of the curvature of the signal in the region where the epoch was expected gave even more disappointing results. The approach finally chosen for this determination is discussed in detail in the next section of this report.

The third phase of the analysis is the actual extraction of the additive components of the signals in Regimes I and III. The signals are broken into polynomial, periodic, and random components by straightforward curve fitting and Fourier techniques. The major assumption in this phase is that the polynomial component, which is a measure of the shift of the data baseline with time, and the periodic function, which is a measure of the vibration of the tube during firing, are additive in their effects on the signal from the induction coil. This implies linearity in measurement and recording equipment as well as independence of the physical effects. The actual approach used in subtracting out the components of the signal is described in the next section and represents the major phase of the analysis presented here.

The final phase of the analysis is the determination of the characteristics of the random "noise" component of the data. The residual signal after both the polynomial and periodic components have been successfully subtracted out, is assumed to be purely random, although its amplitudes are not assumed necessarily to be Gaussian, nor are the amplitudes necessarily uncorrelated in the short term sense. Stated another way, it is not assumed that the power spectrum of the residual noise will be flat or "white".

3. METHODOLOGY

3.1 PROCEDURE.

The first step in the analysis is the identification of the two epochs defining the beginning and end of Regime II. As discussed in Section 2.1, the second regime has certain well-defined features, and these can be exploited in determining the beginning and ending points. This regime begins with a sharp rise in the signal level to a value greatly in excess of the noise level. The signal peaks, then falls slightly before rising to a second somewhat lower peak. Following this second peak, the signal decreases rapidly and may contain one or two short duration spikes. The signal reaches a minimum at the end of Regime II, and moves into Regime III which is essentially a noisy periodic function on a baseline which initially is increasing. The M392 projectiles produce a similar pattern, although the noise between the two peaks is less and there is an absence of spikes during the signal decrease.

A decision must be made as to whether the epochs should be determined solely from this waveform, making no use of any knowledge of the underlying physical phenomena, or whether an attempt should be made to deduce the expected signal from the physical apparatus and use this expected signal as a basis for determining Regime II. Since a better understanding of the physical phenomena is the purpose for which the experiment was performed, it was decided that predicting the waveform in order to determine the regimes was an unjustified bootstrapping, and the determination of epochs was therefore done solely on the basis of the observed waveform. The assumption was made that the waveform consists of two major additive components: a signal consisting of noise, periodic functions, and a growing baseline, to which is added the Regime II signal. There are two characteristics in this overall pattern which are useful in identifying the epochs. The first is the high signal-to-noise ratio on the increasing and decreasing waveforms near the boundaries of Regime II. The second is the high slope of the signal near its first peak. Obscuring the epochs, however, are the noise, which is especially evident in Regimes I and III, the shock signals at the end of Regime II, and the similarities between the periodic waveform at the end of Regime I and the (gradually) increasing slope in the signal at the beginning of Regime II. Any procedure which is used to detect the epochs must take these into account while using the identifying features of the Regime II epochs.

Several approaches to epoch determination were attempted and rejected before the procedure described here was decided upon. The first and simplest was to measure the curvature of the signal near the expected epoch points, and choose that point where the curvature was greatest. With actual signal data, this method was found to be unstable and generally converged to a point which was intuitively unsatisfying. A second approach was to fit polynomials to the data well within Regime II, and well into Regime I, and then extrapolate forward from Regime I and simultaneously extrapolate backwards from Regime II, in the hope that the largest discrepancy between the extrapolated values would occur at the true epoch. Results with this approach were inconsistent, and the computer time required was considered excessive. Methods based directly on differentiating the signal were unsuitable because the differentiating had a markedly deleterious effect on the signal-to-noise ratio, and thus further obscured the epoch which was to be detected.

The algorithm which was actually used to detect the epochs is shown in Figure 3.1. As a reference point from which the first and second epochs are located, the point at which the signal has its maximum first derivative (maximum slope) was used, giving a point just before the peak. (This peak is not necessarily the maximum signal achieved during the shot.) The first epoch is found by a two-step process. Beginning with the reference point, and considering only the preceding two hundred points, points are successively examined until one is found which is less than one of the points preceding it in that interval. This is the initial estimate of the first epoch. The final estimate of the first epoch is determined to be the first point in the interval; bounded by the initial estimate, that begins a monotonically increasing sequence up to the estimate. This epoch is accurate to within about three points in either direction.

The second epoch is found by finding the first occurrence of the minimum whose preceding three points are monotonically decreasing, within a two hundred point interval beginning at the point of the maximum first derivative. Accuracy is estimated to be within one or two points.

Before analysis can begin on the first regime, a secondary epoch detection may be necessary within this subset of the data. This is required when there is a significant change in the fundamental frequency of the periodic component of the signal. The user decides whether or not such a change occurs and, if so, the point of division by subjective analysis of the data set and/or the examination of the power spectra of various subregimes. If this procedure is necessary, there will be three subsets of data, instead of two which must be analyzed separately.

The analysis for the third regime and for the first regime or its two sections are identical. The object is to separate the periodic function, its baseline, and the shocks and noise along the curve. The baseline is assumed to be a low degree polynomial. A fourth degree polynomial was found to be the lowest degree to give a reasonable fit to the curve. The

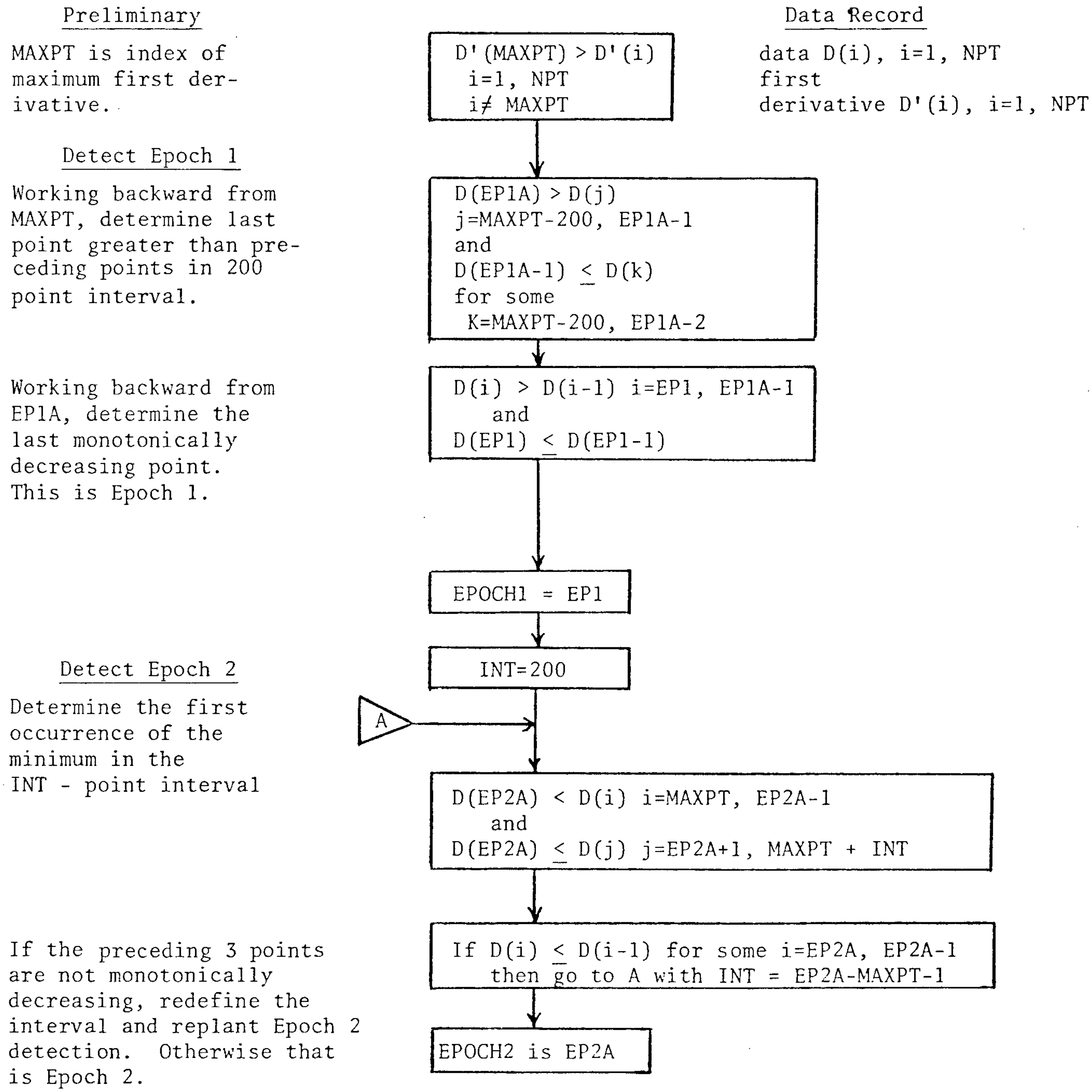


Figure 3.1 Epoch Detection Algorithm

technique used for fitting used a least squares criterion with orthogonalized polynomials. The resulting polynomial was subtracted from the original data, leaving the periodic function and noise. A power spectrum generated for this remainder curve identified the ten frequencies with the most power. These were then used to create a periodic function. The value of ten is somewhat arbitrary but is believed to give a good balance between accuracy and simplicity. The actual residual error cannot be estimated by elementary Fourier techniques because of the presence of the noise. This process is described in more detail in Section 3.2, Basic Analysis Techniques. When the periodic functions and the polynomial have been subtracted from the original curve, the remaining signal consists primarily of noise, with some shocks. When the periodic function is added to the polynomial, the result is a good approximation to the original noiseless signal.

In the final step of the analysis, the noise in the first and third regimes is examined. The power spectrum for each is generated. The noise is then characterized as white noise or, if it is non-white, its dominant frequencies are specified. A χ^2 test on the noise gives the mean and standard deviation of the noise and indicates the probability of the noise being Gaussian.

Identification of the shocks remaining in the noise is done by determining the points greater than 2.5 standard deviations from the mean. These are printed and points which are more than 3 standard deviations from the mean are specially noted. In addition, the number of standard deviations of the first derivative from its mean are printed for those points. It is expected that a shock occurs when both measures give high deviations and/or when adjacent points are far from the mean in opposite directions.

This concludes the analysis of the gun signal.

3.2 BASIC ANALYSIS TECHNIQUES.

Most of the analysis techniques required for the inductance coil data are standard mathematical and statistical procedures. A few techniques, including those for the detection of damaged data and for epoch detection, are peculiar to this particular problem, and these required the development of specialized computer subroutines. In this section, techniques and subroutines acquired by Falcon for the coil data analysis will be discussed. Not all of the techniques described here are currently used in the data analysis, but they have been included because of their past or potential usefulness in related work.

Three of these techniques are taken directly from the collection of subroutines written and issued by the International Mathematical and Statistical Libraries, Inc. This collection, known as the "IMSL" library, is available in a number of data processing centers throughout the country. The BRL CYBER computer center currently subscribes to the IMSL library. Due to the large data arrays, it was necessary to create a special library of IMSL routines for MOOSE, in which subroutine storage arrays were increased and stored in Level 2. All IMSL routines needed by MOOSE were in this IMSL2 library.

The first basic technique concerns time series analysis in the time and frequency domains. In particular, a power spectrum representation for a given time series is needed to identify the dominant frequencies in the data set. This is accomplished by the IMSL subroutine FTFREQ. For a given time series, this subroutine computes the mean and variance, the autocovariance function, and the power spectrum. The resolution of this subroutine is 250 Hz when the time series contains at least 800 points taken at 2.5 microsecond intervals, decreasing to 1 kHz resolution when only 200 points are available.

Given the power spectrum of the periodic component of the data, a "noise free" approximation of this component was synthesized from ten sinusoids. The frequencies for these sinusoids were selected to be those ten frequencies for which the power spectrum showed the greatest values. The phases and amplitudes for the sinusoidal components were determined by the following process:

x_j = point in original curve $\quad j = 1, NPT$ (# points)

f_i = desired frequency $\quad i = 1, NFR$ (# frequencies)

AC_i, BS_i = coefficients of cosine and sine components of f_i

$$AC_i = 2 \sum_{j=1}^{NPT} x_j \cos\,(2\pi (j-1) f_i \Delta t)$$

$$BS_i = 2 \sum_{j=1}^{NPT} x_j \sin\,(2\pi (j-1) f_i \Delta t)$$

PER_j = approximation curve $\quad j = 1, NPT$

$$PER_j = \text{mean }(x) + \frac{1}{NPT} \sum_{i=1}^{NFR} \left\{ \begin{matrix} AC_i \cos\,(2\pi (j-1) f_i \Delta t) + \\ BS_i \sin\,(2\pi (j-1) f_i \Delta t) \end{matrix} \right\}$$

The technique for fitting a polynomial to the data set, in order to obtain a "noise free" approximation to the baseline, was selected directly from the IMSL library. The subroutine chosen was RLFOR, and uses orthogonal polynomials in a univariate curvilinear regression model to fit a data set. The user specifies the maximum degree for the polynomial, the weight of each point, and a limiting value for the measure of goodness of fit. This limiting value defines the degree of the polynomial. The output includes the coefficients of the fitted polynomial and the values of the polynomial function for the data set.

The third IMSL subroutine is GFIT, which tests the goodness of fit of a data set to a particular distribution. The χ^2 test is performed using the number of cells specified by the user. The distribution used for comparison is currently the Gaussian distribution. The pertinent output is the χ^2 statistic and the Q-statistic, which is the probability that χ^2 is exceeded if the data do not fit the distribution. In other words, a small Q implies that the data fit the distribution, while a large Q implies that they do not.

Before the analysis of a data set can begin, it must be determined whether the data are suitable for complete analysis, or have been damaged through instrumentation saturation to the extent that only partial analysis can be accomplished. The data set classification checks for multiple occurrences of the maximum signal. There are three classes of data sets: no saturation, limited saturation with less than ten points at the maximum value, and heavy saturation with at least ten points at the maximum value. The subroutine which makes this determination was programmed by Falcon using an algorithm supplied by BRL.

Several data-smoothing options exist. These include the 11-point moving average and the 11-point moving median. These smoothing techniques are not needed, nor are they used, in the current analysis since the polynomial and sinusoidal fitting techniques inherently smooth the data.

The calculations of the first and second derivatives are accomplished by means of central differences. The equations are:

$$f'(x) = \frac{1}{12h} (8f(x+h) - 8f(x-h) - f(x+2h) + f(x-2h)) \tag{1}$$

$$f''(x) = \frac{1}{12h^2} (16f(x+h) + 16f(x-h) - 30f(x) - f(x+2h) - f(x-2h)) \tag{2}$$

In addition, routines exist which take the mean, median, or standard deviation of a set of points. One routine sorts the rows of two column vectors so that the second column is in descending order.

Finally, it is possible to choose a subset of the data set by specifying the beginning and ending times of the subset and the interval between the points in the data set.

3.3 PLOTTING CAPABILITIES.

Without question the most efficient means for presenting the results of the analysis of a large data set is through graphs and charts. A salient feature of the work described here is the generation of plots at a number of stages of the analysis. For this purpose, the CALCOMP Model 780 digital plotting system was used. This consists of a model 770 tape system and a model 763 plotter. Two packages of plotting subroutines are used: the CALCOMP plotting routines for the BRL, CDC computing system, and the BRLESC FORTRAN plotting subroutines[5]. With these plotting packages, graphs can be plotted in a variety of ways. One or more graphs can be plotted on one page using axes, borders, or grids, with either linear or logarithmic spacing. The points of the graph can be represented by any one of a large number of symbols and can be connected, if desired by solid or broken lines. Titles and headings can be included on the plot. A major advantage of this plotting system is that the page can be as large as 29 inches by 120 feet, so that the resolution possible is limited only by the 2.5 microsecond data sampling rate and not by the plotter itself.

There are currently four plotting options which can be utilized. The first is the power spectrum for the first 25 frequencies, plotted on a semilog graph, with $\log_{10}$ power plotted against frequency. The second option is a graph of the entire set of raw data, along with the first and second derivatives of the data. The third option shows the analysis of a subset of the data, and consists of three pages. The first page contains a plot of the raw data in a solid line and a broken line for the polynomial fit. The second page gives the raw data minus the polynomial as a solid line, and the per-iodic fit as a broken line. The third page contains only the noise, which is the polynomial and periodic fits subtracted from the raw data. The fourth plotting option shows the entire data set with the noise subtracted from the first and third regimes.

---

[5] *Monte W. Coleman, John V. Lanahan, "BRLESC FORTRAN Plotting Subroutines," ARDC-TR-6, July 1970, AD 711893.*

4. MOOSE PROGRAM

The MOOSE (MOdulated OScillator Evaluation) Program is listed in Appendix A.1. The program was written in FORTRAN IV, compatible with the BRL CYBER computer system, using structured programming techniques. The CALCOMP plotting library is required for this program along with IMSL2, a special library of IMSL routines, which have been converted to level 2 storage. Appendix A.2 lists the necessary IMSL routines which are included in the IMSL2 library.

The MOOSE program uses level 2 storage and organizes subroutines into different overlays for loading because of the size of the program and the storage required for it. Figure 4.1 shows the overall flow of the program. Figure 4.2 shows the organization of the overlays used in the program. The size and storage requirements mean that the program can only be run on mainframe "Z". The program in UPDATE form as MOOSEOLDPL is located on mainframe "A". The inductance coil data is in a file on mainframe "Z" and is attached as TAPE1.

The job control language for running the MOOSE program is shown in Appendix B.1. Appendix B.2 gives a description of the input data and Appendix B.3 shows an input sample. The output description of the program is listed in Appendix C.1 and Appendix C.2 contains an output sample. Execution error codes are listed in Appendix C.3.

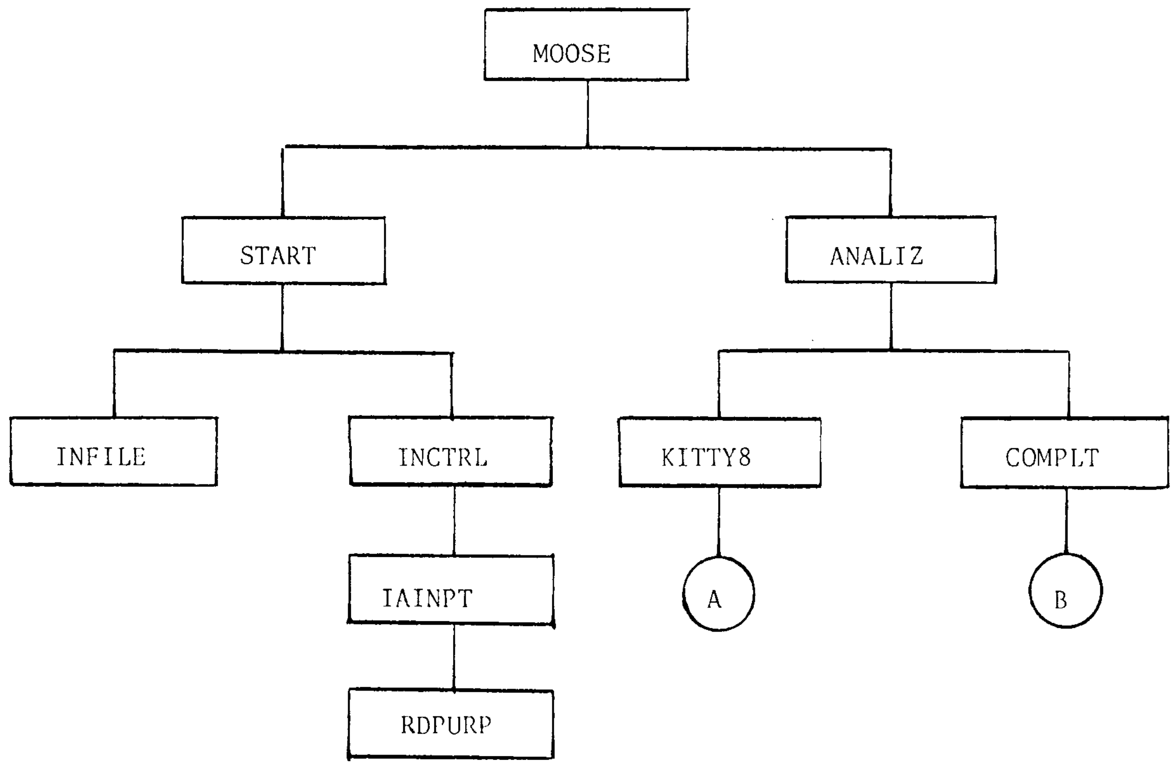


Figure 4.1 MOOSE Program Flowchart

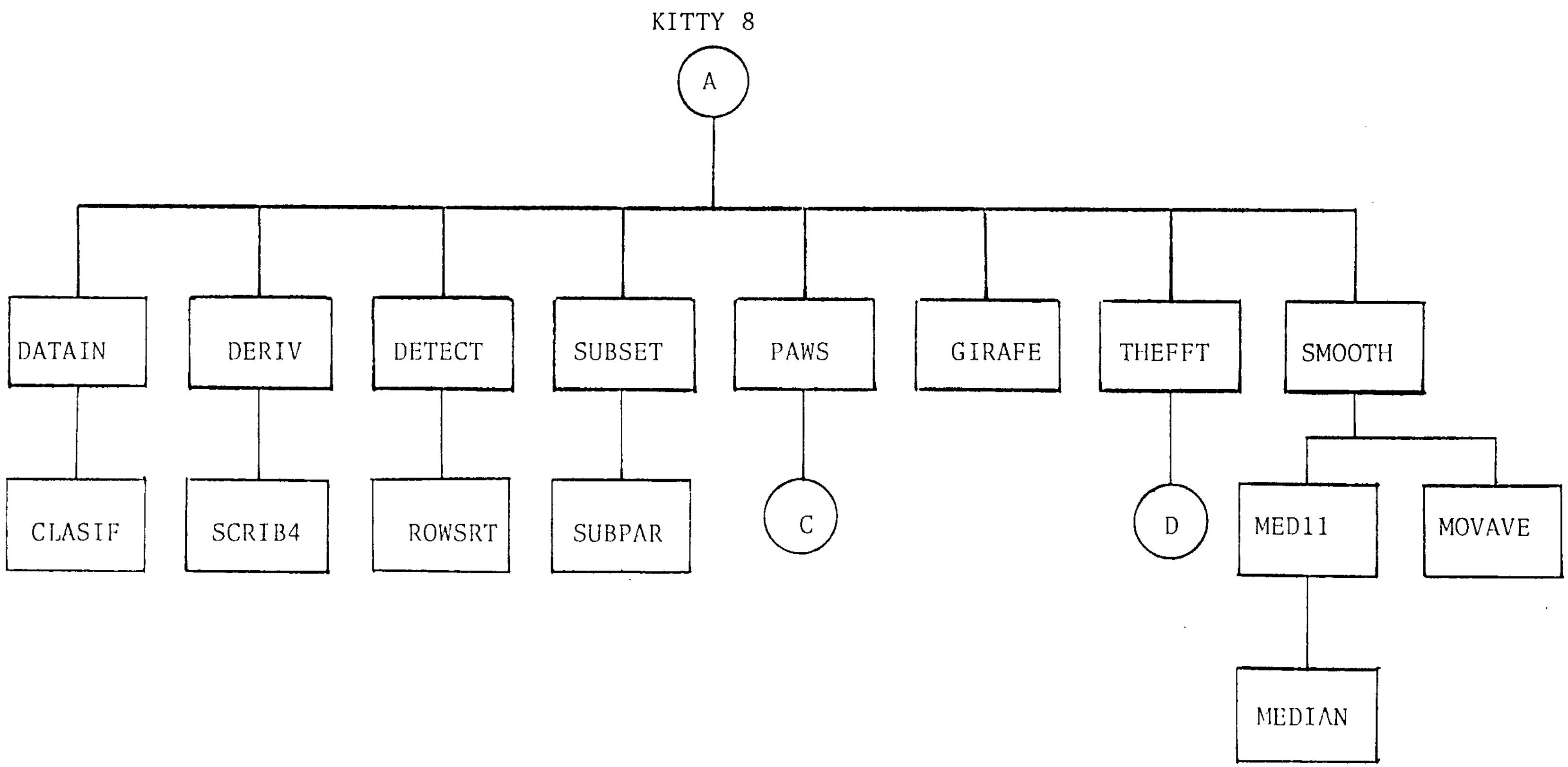


Figure 4.1 MOOSE Program Flowchart (Continued)

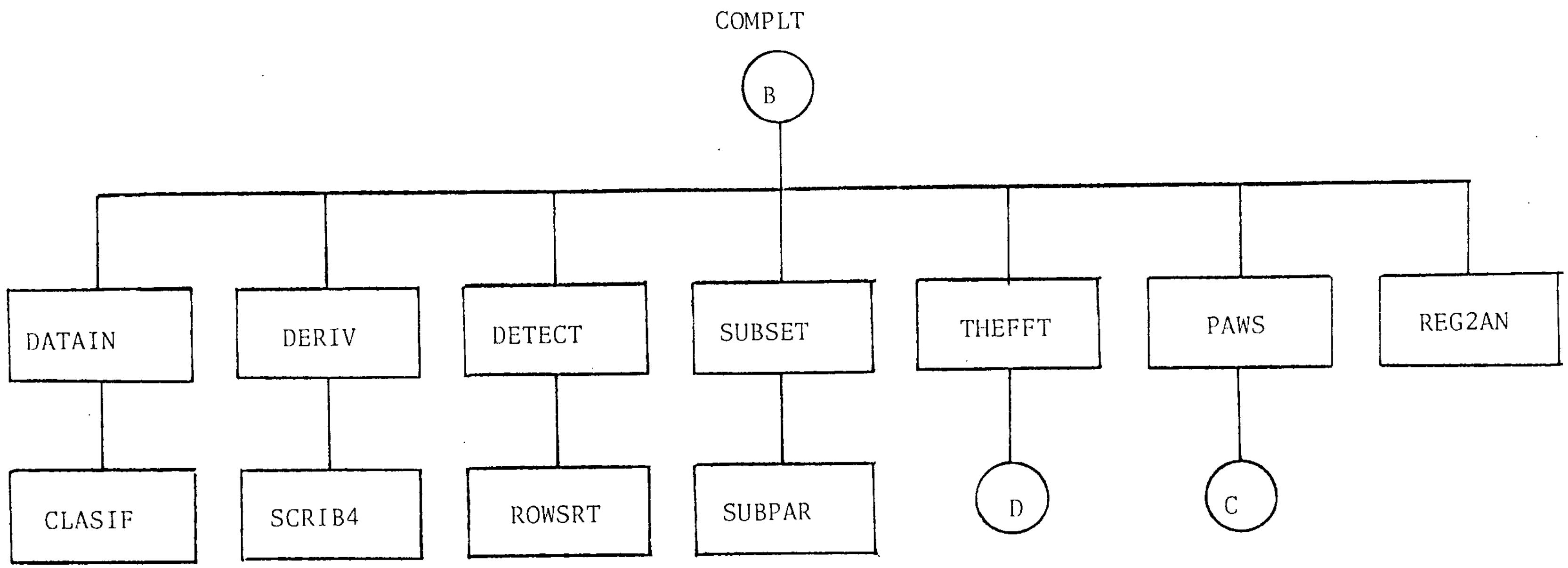


Figure 4.1 MOOSE Program Flowchart (Continued)

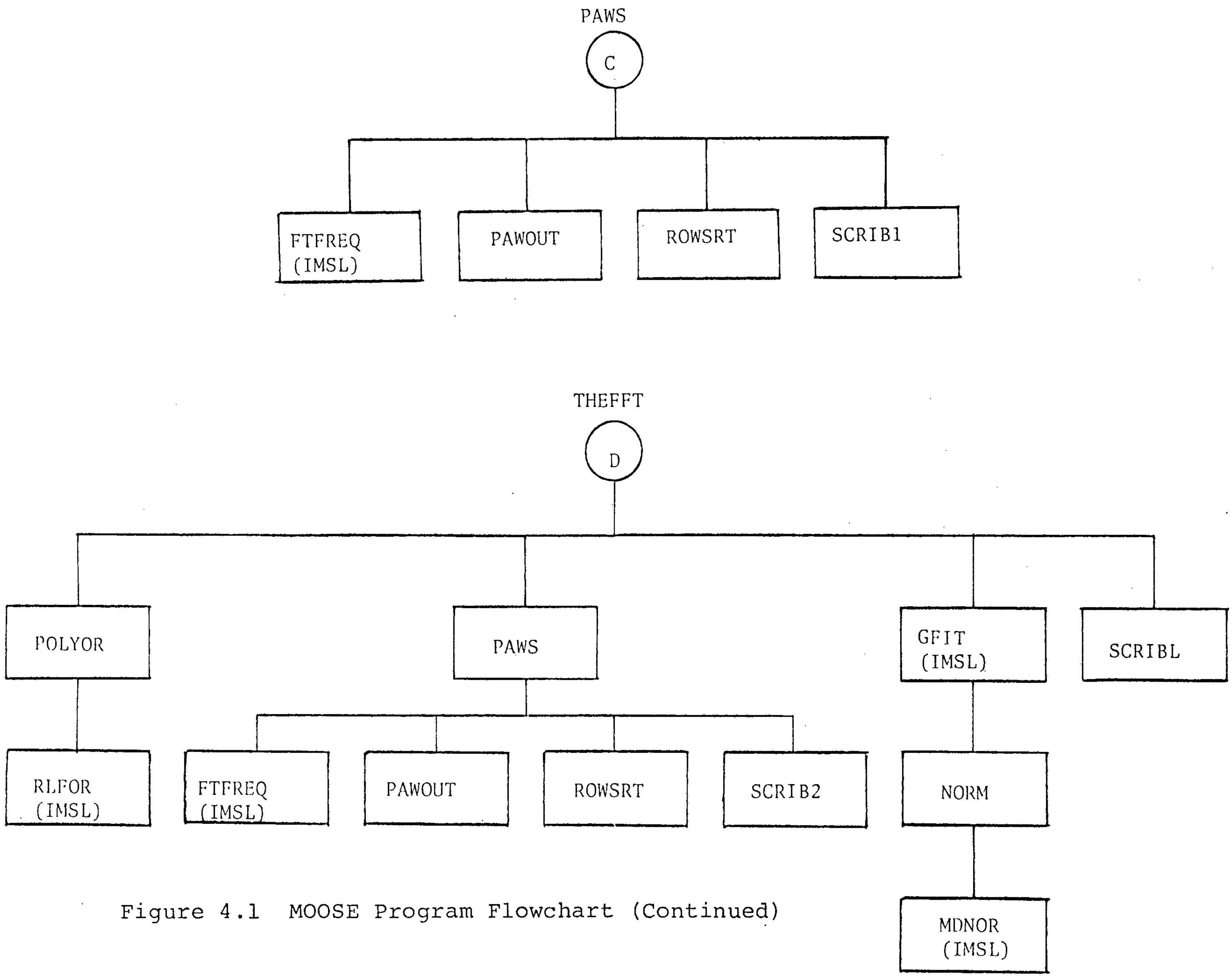


Figure 4.1 MOOSE Program Flowchart (Continued)

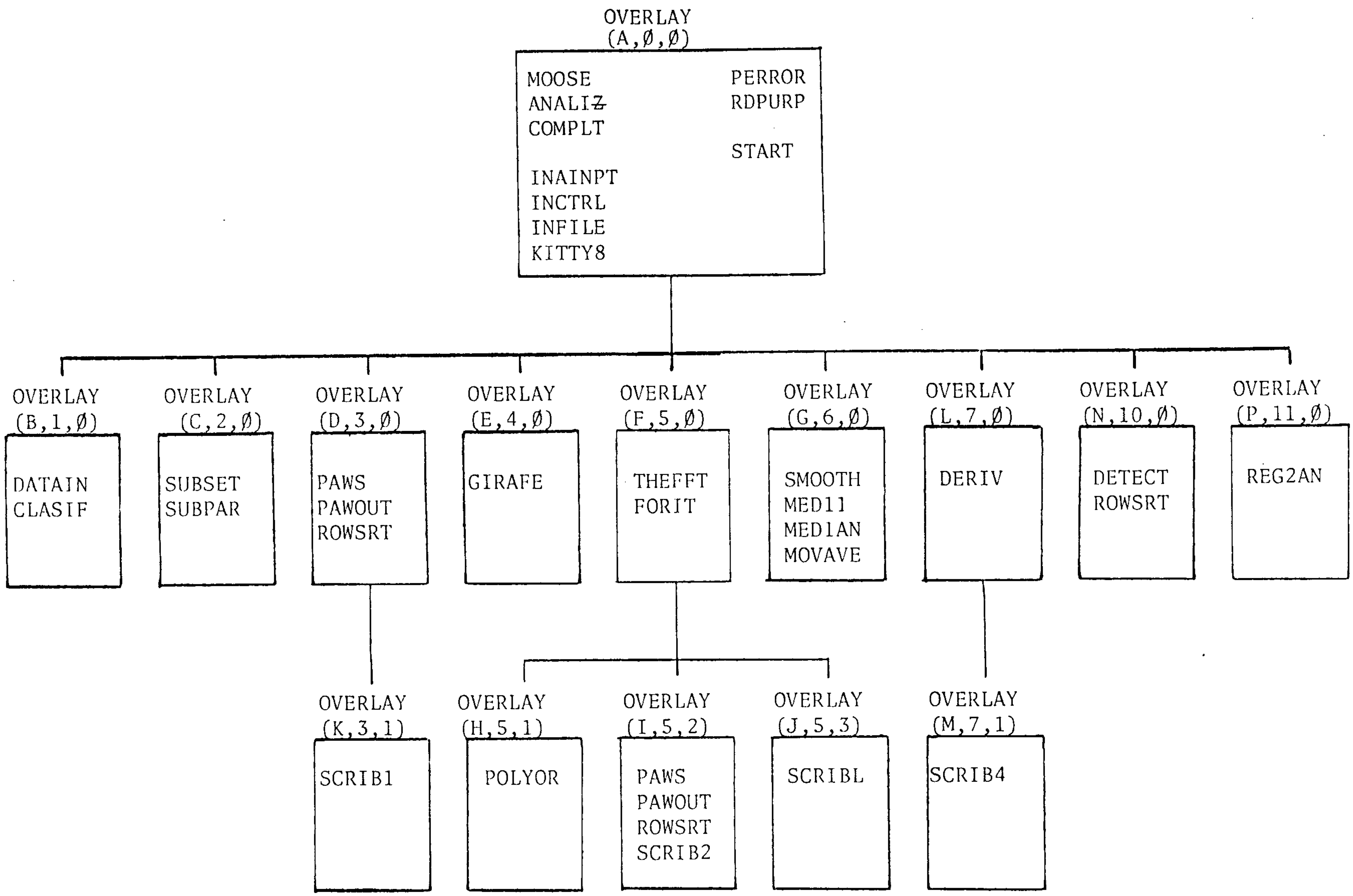


Figure 4.2 MOOSE Program Overlay Organization

REFERENCES

1. R. K. Loder, J. O. Pilcher, "Nondestructive Test Method to Establish the Performance of Projectile-Gun Systems," Proceedings of the 26-th Defense Conference on Nondestructive Testing, 15 to 17 November 1977, Seattle, Washington; Published by U.S. Army Materials and Mechanics Research Center, Watertown, MA (pages 272-303).

2. R. K. Loder, J. Q. Schmidt, "Radio Frequency Oscillator Technique for Monitoring Velocity and Structural Integrity of Projectiles from Their Exit from the Muzzle," ARBRL-MR-03100, USA ARRADCOM, Ballistic Research Laboratory, Aberdeen Proving Ground, MD, April 1981. (AD A100725)

3. G. Schultze, "Telemetrische V_o-Messungen"; Colloquium on Ballistic Measurement Techniques; held at the Institut Franco-Allemand De Recherches De Saint-Louis, 6 and 7 June 1973; ISL Report 14/73.

4. J. Q. Schmidt, "A Radio Frequency Oscillator Technique for Measuring Projectile Muzzle Velocity," USARRADCOM, Ballistic Research Laboratory, Aberdeen Proving Ground, MD; Technical Report ARBRL-TR-02158, April 1979. (AD B038926L)

5. Monte W. Coleman, John V. Lanahan, "BRLESC FORTRAN Plotting Subroutines," ARDC-TR-6, July 1970, AD 711 893.

APPENDIX A

THE COMPUTER PROGRAM

APPENDIX A.1

TABULATION

```
C-----------------------------------------------------------------------
C                      USER DIAGNOSTICS
CLOCATION   CODE MODE/RESULT         USER DIAGNOSTIC MESSAGE
C--------- ---- -----------    ---------------------------------------
CSR-SELECT   2   BATCH/FATAL    IMPROPER SELECTION MADE FOR PROCESSING
CSR-SELECT   2   IA    /INF     OPTION. CURRENTLY ALLOWED OPTIONS ARE:
CSR-SELECT   2                            END       -TO END PROCESSING
CSR-SELECT   2                            ANALYSIS -TO PERFORM AN ANALYSIS
CSR-SELECT   1   BATCH/FATAL    IMPROPER SELECTION MADE FOR RUN MODE.
CSR-SELECT   1   IA    /FATAL   PROPER SELECTIONS ARE:
CSR-SELECT   1                            0         - FOR INTERACTIVE MODE
CSR-SELECT   1                            1         - FOR BATCH MODE
CSR-SELECT   1                  FOR THE VARIABLE IFBACH
CSR-ANALIZ   4   BATCH/FATAL    IMPROPER SELECTION MADE FOR ANALYSIS
CSR-ANALIZ   4   IA    /NA      OPTION. CURRENTLY ALLOWED OPTIONS ARE:
CSR-ANALIZ   4                            0         - FOR EITHER MODE
CSR-ANALIZ   4                  FOR THE VARIABLE IANOPT
CSR-ANALIZ   5   BATCH/NA       IMPROPER SELECTION MADE FOR ANALYSIS
CSR-ANALIZ   5   IA    /INF     OPTION. CURRENTLY ALLOWED OPTIONS ARE:
CSR-ANALIZ   5                            0         - FOR EITHER MODE
CSR-ANALIZ   5                  FOR THE VARIABLE IANOPT
      OVERLAY(A,0,0)
      PROGRAM MOOSE (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE1,TAPE2,
     1               TAPE3,TAPE10,DEBUG=OUTPUT)
C-----------------------------------------------------------------------
C
C * THIS PROGRAM PERFORMS ONE OR MORE USER SELECTED TRANSFORMATIONS
C *  ON GUN-PROJECTILE DYNAMICS DATA.
C
C * FILE HANDLING INFORMATION:  TAPE5 - USER INPUTS ON "INPUT"
C *                             TAPE6 - RUN TIME OUTPUT SUMMARY
C *                             TAPE1 - GUN-PROJECTILE DATA FROM PF
C *                             TAPE2 - LOCAL FILE TO HOLD RUN INFO
C *                             TAPE3 - OUTPUT FROM DATA TRANSFORMATION
C *                                     WHICH USER MAY COPY TO OUTPUT
C *                                               OR
C *                                     CATALOG FOR RETENTION
C *                            TAPE10 - USED FOR CALCOMP PLOTTING
C
C------------------------------------------FALCON R&D  JLW  4/19/79
      COMMON / SSDATA / NSSPTS , SDTIME , SUBSTR(5000) , SSTIME(5000)
      COMMON / SDATA / NSTPTS , DTIME , STRING(5000) , STIME(5000)
      COMMON / FREQS / FREQEN(20) , NFR
      COMMON /REPDAT / REPSTR(5000) , IFREP , NORSTR(5000)
      COMMON / LABEL / KTB(24) , TSTART , I
      COMMON / DERIVS / STRDOT(5000) , STRDD(5000) , PLOTOP
      COMMON / CLASCM / ICLASS , NBEGBL
      LEVEL 2, NSSPTS, SDTIME, SUBSTR, SSTIME, NSTPTS, DTIME,
     A         STRING, STIME, FREQEN, NFR,
     B         REPSTR,IFREP,NORSTR,KTB,TSTART,I,
     C         STRDOT,STRDD,PLOTOP,ICLASS,NBEGBL
C
C * WRITE HEADING FOR MOOSE
C
      WRITE(6,9001)
C
C * START UP PROCESSING.
```

PROGRAM LISTING

```
C
      CALL START
C
C * SELECT USER OPTION FOR DATA TRANSFORMATION DESIRED.
C
      CALL ANALIZ
C
C * FINISH RUN
C
C
 9000 STOP "* NORMAL TERMINATION *"
 9001 FORMAT(1H1,///,20X,54(1H*),/,20X,1H*,52X,1H*,/,20X,1H*,
     *        22X,9HM O O S E,21X,1H*,/,20X,1H*,52X,1H*,/,
     *        20X,54(1H*))
      END
```

PROGRAM LISTING (Continued)

```
C-----------------------------------------------------------------------
C
C * THIS BLOCK DATA SUBPROGRAM SETS INITIAL DATA VALUES AT DEFAULT LEVEL
C
C-----------------------------------------------FALCON R&D   JLW   4/20/79
      BLOCK DATA INITAL
C
C * FILE RELATED DATA
C
C
C * PROGRAM CONTROL VARIABLES
C
      COMMON / CONTRL / LUIN , LUOUT , LUDATA , LURUN , LUREPT
C
      COMMON / CONTRL / IFSVRP , ONLINE , IANOPT , OPTION , IFBACH
      COMMON / CONTRL / USRNAM , PURPOS
C
C * TYPE DECLARATION
C
      INTEGER ONLINE , OPTION , USRNAM(2) ,PURPOS(6)
      LEVEL 2, LUIN,LUOUT,LUDATA,LURUN,LUREPT,IFSVRP,ONLINE,IANOPT
      LEVEL 2, OPTION,IFBACH,USRNAM,PURPOS
      DATA LUIN,LUOUT,LUDATA,LURUN,LUREPT / 5, 6, 1, 2, 3 /
C
C * SET CONTROL DEFAULTS.
C
      DATA IFSVRP / 0 /
      DATA ONLINE /0 /
      END
```

PROGRAM LISTING (Continued)

```
      SUBROUTINE ANALIZ
C-----------------------------------------------------------------------
C
C * THIS ROUTINE CONTROLS PROCESSING OF ALL ANALYSIS OPTIONS
C
C---------------------------------------------FALCON R&D  JLW  4/20/79
C
C * SELECT AN ANALYSIS OPTION
C
C
C * PROGRAM CONTROL VARIABLES
C
      COMMON / CONTRL / LUIN , LUOUT , LUDATA , LURUN , LUREPT
C
      COMMON / CONTRL / IFSVRP , ONLINE , IANOPT , OPTION , IFBACH
      COMMON / CONTRL / USRNAM , PURPOS
C
C * TYPE DECLARATION
C
      INTEGER ONLINE , OPTION , USRNAM(2) ,PURPOS(6)
      LEVEL 2, LUIN,LUOUT,LUDATA,LURUN,LUREPT,IFSVRP,ONLINE,IANOPT
      LEVEL 2, OPTION,IFBACH,USRNAM,PURPOS
C
      READ(LUIN,9002) IANOPT
      WRITE(LUOUT,9003) IANOPT
C
      IF (IANOPT .EQ. 0) GO TO 9000
      IF(IANOPT.EQ.3)CALL SKPDAT
      IF (IANOPT .EQ. 1) CALL KITTY8
      IF (IANOPT .EQ. 2) CALL COMPLT
      IF (IANOPT .LT. 0) CALL PERROR(5)
      IF(IANOPT.GT.3)CALL PERROR(8)
C
 9000 RETURN
C
 9001 FORMAT(20X,40HSELECT ANALYSIS OPTION 0, 1, OR 2 NOW.. /)
 9002 FORMAT(I2)
 9003 FORMAT(1H1,///,10X,18HANALYSIS OPTION IS,I5,
     *        /,15X,23H0 = NO FURTHER ANALYSIS,/,
     *        15X,28H1 = USER CONTROLLED ANALYSIS,
     *        /,15X,33H2 = COMPLETE DATA RECORD ANALYSIS,
     *        /,15X,13H3 = SKIP DATA,
     *        //)
C
      END
```

PROGRAM LISTING (Continued)

```
      SUBROUTINE COMPLT
C-------------------------------------------------------------
C
C * SUBROUTINE COMPLT RUNS A COMPLETE SPECTRUM OF
C       TRANSFORMATIONS ON GUN-PROJECTILE DYNAMICS DATA
C
C-------------------------------------------------------------
      COMMON / COMSUB / TINIT , TLAST , NODTS
      COMMON / SDATA  / NSTPTS , DTIME , STRING(5000) , STIME(5000)
      COMMON / DTECT  / EPOCH1 , EPOCH2
      COMMON / REPDAT / REPSTR(5000) , IFREP  , NDREP
      COMMON / CLASCM / ICLASS , NBEGBL
      COMMON / REGII  / MAXPT, MINPT, VALMAX, VALMIN
      COMMON / CMPLT  / TEND1 , TEND2 , IDIV , SUBTIM
     *                 ,TLEN1 , TLEN2 , TLEN
      INTEGER EPOCH1 , EPOCH2
      LEVEL 2, TINIT,TLAST,NODTS,NSTPTS,DTIME,STRING,STIME,EPOCH1,EPOCH2
     *        , ICLASS,NBEGBL,TEND1,TEND2,IDIV,SUBTIM,NDREP
     *        , REPSTR,IFREP,TLEN1,TLEN2,TLEN
     *        , MAXPT,MINPT,VALMAX,VALMIN
C**********************************************************************
C
C * READ IN COMMON DATA
C       AND DETERMINE SATURATION LEVEL
C
      CALL OVERLAY(1HB,1,0)
C
C * CALCULATE THE FIRST AND SECOND DERIVATIVES FOR
C     UNSATURATION PORTION OF DATASET
C
      CALL OVERLAY(1HL,7,0)
C
C * LOCATE EPOCHS 1 AND 2 OR EPOCH 1 IF SATURATED DATA SET
C
      CALL OVERLAY(1HN,8,0)
C**********************************************************************
C
C * WRITE PAGE HEADING FOR COMPLT
C
      WRITE(6,9001)
C
C * ANALYSIS OVERLAYS ARE:
C        (C,2,0)  SUBSET
C        (F,5,0)  THEFFT
C        (D,3,0)  PAWS
C        (P,9,0)  REG2AN
C
C * CHOOSE SUBSET FOR REGIME ONE
C
      TINIT = STIME(1)
      TLAST = STIME(EPOCH1)
      NODTS = 1
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C * CAN REGIME ONE BE SUBDIVIDED?
C     TEND1 = SMALLEST ALLOWABLE TIME FOR SUBDIVISION OF REGIME I
C     TEND2 = LARGEST ALLOWABLE TIME FOR SUBDIVISION OF REGIME I
```

PROGRAM LISTING (Continued)

```
C
      TEND1 = TINIT + 201. * DTIME
      TEND2 = TLAST - 201. * DTIME
      IF (TEND1 .GE. TEND2) GO TO 1000
 3000 CONTINUE
C
C * USER DECIDES IF REGIME ONE NEEDS TO BE SUBDIVIDED
C
      WRITE(6,9002) TINIT , TLAST
      READ (5,9003) IDIV
C
C * IDIV = 1, ANALYSIS CONTROL RETURNS TO USER
C        = 0, REGIME I WILL NOT BE SUBDIVIDED
C        = -1, REGIME I WILL BE DIVIDED BY THE USER
C
      IF (IDIV .EQ. 1) GO TO 9000
      IF (IDIV .EQ. 0) GO TO 1000
      IF (IDIV .EQ. -1) GO TO 2000
      CALL PERROR(71)
 2000 CONTINUE
C
C * SUBTIME IS TIME OF SUBDIVISION OF REGIME I
C
      READ (5,9005) SUBTIM
C
C * CHECK VALIDITY OF SUBTIME
C
      IF (SUBTIM .LE. TINIT .OR. SUBTIM .GE. TLAST)
     *              CALL PERROR(73)
      IF (SUBTIM .GT. TINIT .AND. SUBTIM .LE. TEND1)
     *  SUBTIM = TEND1 + DTIME
      IF (SUBTIM .GE. TEND2 .AND. SUBTIM .LT. TLAST)
     *  SUBTIM = TEND2 - DTIME
      WRITE(6,9004) TEND1 , TEND2 , SUBTIM
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C * THREE ANALYSES OF DATA - TWO SUBDIVISIONS OF REGIME I
C
      NDREP = 3
C
C * WRITE HEADING FOR REGIME I PART ONE
C
      IP = 1
      TLAST = SUBTIM
      TLEN1 = TLAST - TINIT
      WRITE(6,9006) IP, TINIT,TLAST,TLEN1
C
C * SELECT PART ONE OF REGIME ONE
C
      CALL OVERLAY(1HC,2,0)
C
C * ANALYZE PART ONE OF REGIME ONE
C
      CALL OVERLAY(1HF,5,0)
      CALL OVERLAY(1HD,3,0)
C
C * WRITE HEADING FOR REGIME I - PART TWO
```

PROGRAM LISTING (Continued)

```
C
      IP = 2
      TINIT = SUBTIM
      TLAST = STIME(EPOCH1)
      TLEN2 = TLAST - TINIT
      WRITE(6,9006) IP, TINIT, TLAST, TLEN2
C
C * SELECT PART TWO OF REGIME ONE
C
      CALL OVERLAY(1HC,2,0)
C
C * ANALYZE PART TWO OF REGIME ONE
C
      CALL OVERLAY(1HF,5,0)
      CALL OVERLAY(1HD,3,0)
      GO TO 4000
 1000 CONTINUE
C
C * TWO ANALYSES OF DATA
C
      NDREP = 2
C
C * CHOOSE SUBSET FOR REGIME ONE
C
      CALL OVERLAY(1HC,2,0)
C
C * WRITE HEADING FOR REGIME I
C
      TLEN = TLAST - TINIT
      WRITE(6,9007) TINIT , TLAST , TLEN
C
C * RUN THEFFT AND PAWS ON REGIME ONE
C
      CALL OVERLAY(1HF,5,0)
      CALL OVERLAY(1HD,3,0)
 4000 CONTINUE
C
C * ICLASS INDICATES SATURATION LEVEL:
C       ICLASS = -1, HEAVY SATURATION, EPOCH 1 DETECTION
C                    AND REGIME 1 ANALYSIS.
C       ICLASS =  0, NO SATURATION, FULL ANALYSIS
C       ICLASS =  1, LIMITED SATURATION (CLIPPING),
C                    FULL ANALYSIS.
C
      IF (ICLASS .EQ. -1) GO TO 9000
C***********************************************************************
C
C * ANALYSIS OF REGIME II
C
C * CHOOSE A SUBSET
C
      TINIT = STIME(EPOCH1)
      TLAST = STIME(EPOCH2)
      NODTS = 1
      CALL OVERLAY(1HC,2,0)
C
C * WRITE HEADING FOR REGIME II
```

PROGRAM LISTING (Continued)

```
C
      TLEN = TLAST - TINIT
      WRITE(6,9008) TINIT, TLAST, TLEN
C
C * SPECIAL REGIME II ANALYSIS
C
      CALL OVERLAY(1HP,9,0)
C
      WRITE(6,9010) MAXPT, VALMAX, MINPT, VALMIN
C
C * POWER SPECTRUM
C
      CALL OVERLAY(1HD,3,0)
C***********************************************************************
C
C * CHOOSE SUBSET FOR REGIME THREE
C
      TINIT = STIME(EPOCH2)
      TLAST = STIME(NSTPTS)
      NODTS = 1
      CALL OVERLAY(1HC,2,0)
C
C * WRITE HEADING FOR REGIME III
C
      TLEN = TLAST - TINIT
      WRITE(6,9009) TINIT, TLAST, TLEN
C
C * RUN THEFFT AND PAWS ON REGIME THREE
C
      CALL OVERLAY(1HF,5,0)
      CALL OVERLAY(1HD,3,0)
C***********************************************************************
 9000 CONTINUE
 9001 FORMAT(1H1,///,20X,54(1H*),/,20X,1H*,52X,1H*,/,20X,
     *        1H*,9X,34HANALYSIS OF REGIMES I, II, AND III,9X,
     *        1H*,/,20X,1H*,52X,1H*,/,20X,54(1H*))
 9002 FORMAT(1X,////,10X,38HREGIME ONE IS DEFINED TO START AT TIME,
     *        F8.4,2X,15HAND END AT TIME,F8.4,2X,1H.,/,10X,
     *        50HIF IT HAS TWO DISTINCT SECTIONS TO IT, THEN IT IS ,
     *        31HBEST TO DIVIDE IT IN TWO PARTS.)
 9003 FORMAT(I2)
 9004 FORMAT(1X,///,10X,44HTHE OPTION TO SUBDIVIDE REGIME ONE HAS BEEN ,
     *        8HCHOOSEN.,/,10X,
     *        45HTHE POINT OF SUBDIVISION  MAY BE BETWEEN TIME,
     *        F8.4,9H AND TIME,F8.4,/,10X,
     *        35HTHE POINT OF SUBDIVISION IS AT TIME,F8.4,2X,1H.)
 9005 FORMAT(F8.4)
 9006 FORMAT(1H1,////,20X,30(1H*),/,20X,1H*,28X,1H*,/,20X,
     *        1H*,5X,17HREGIME I ANALYSIS,6X,1H*,/,
     *        20X,1H*,28X,1H*,/,20X,30(1H*),///,20X,4HPART,I2,/,
     *        30X,F8.4,2X,15HMILLISECONDS TO,F8.4,2X,
     *        12HMILLISECONDS,/,30X,F8.4,2X,17HMILLISECONDS LONG)
 9007 FORMAT(1H1,////,20X,30(1H*),/,20X,1H*,28X,1H*,/,
     *        20X,1H*,5X,17HREGIME I ANALYSIS,6X,1H*,/,
     *        20X,1H*,28X,1H*,/,20X,30(1H*),///,30X,
     *        F8.4,2X,15HMILLISECONDS TO,F8.4,2X,
     *        12HMILLISECONDS,/,30X,F8.4,2X,17HMILLISECONDS LONG)
```

PROGRAM LISTING (Continued)

```
9008 FORMAT(1H1,////,20X,30(1H*),/,20X,1H*,28X,1H*,/,20X,1H*,5X,
    *         18HREGIME II ANALYSIS,5X,1H*,/,20X,1H*,28X,1H*,/,20X,
    *         30(1H*),///,30X,F8.4,2X,15HMILLISECONDS TO,F8.4,2X,
    *         12HMILLISECONDS,/,30X,F8.4,2X,17HMILLISECONDS LONG)
9009 FORMAT(1H1,////,20X,30(1H*),/,20X,1H*,28X,1H*,/,20X,1H*,5X,
    *         19HREGIME III ANALYSIS,4X,1H*,/,20X,1H*,28X,1H*,/,20X,
    *         30(1H*),///,30X,F8.4,2X,15HMILLISECONDS TO,F8.4,2X,
    *         12HMILLISECONDS,/,30X,F8.4,2X,17HMILLISECONDS LONG)
9010 FORMAT(1X,////,10X,25HMAXIMUM ACHIEVED AT POINT,
    *         I10,2X,10HWITH VALUE,F10.4,/,10X,
    *         25HMINIMUM ACHIEVED AT POINT,I10,2X,
    *         10HWITH VALUE,F10.4,///)
      RETURN
      END
```

PROGRAM LISTING (Continued)

```
      SUBROUTINE IAINPT
C-----------------------------------------------------------------------
C
C * THIS ROUTINE PROMPTS INTERACTIVE INPUT FROM THE USER.
C
C--------------------------------------------------FALCON R&D JLW 6/5/79
C
C
C * PROGRAM CONTROL VARIABLES
C
      COMMON / CONTRL / LUIN , LUOUT , LUDATA , LURUN , LUREPT
C
      COMMON / CONTRL / IFSVRP , ONLINE , IANOPT , OPTION , IFBACH
      COMMON / CONTRL / USRNAM , PURPOS
C
C * TYPE DECLARATION
C
      INTEGER ONLINE , OPTION , USRNAM(2) ,PURPOS(6)
      LEVEL 2, LUIN,LUOUT,LURUN,LUREPT,LUDATA,IFSVRP,ONLINE,IANOPT
      LEVEL 2, OPTION,IFBACH,USRNAM,PURPOS
C
C * PROMPT USER NAME INPUT
C
      WRITE(LUOUT,9001)
C
      READ(LUIN,*) USRNAM
      WRITE(LUOUT,9005) USRNAM
C
C * PROMPT TEXT TO DOCUMENT PURPOSE OF THE RUN.
C
      WRITE(LUOUT,9002)
C
      CALL RDPURP
C
 9000 RETURN
C
 9001 FORMAT(1X,////,10X,11HUSER'S NAME)
 9002 FORMAT(1X,///,10X,14HPURPOSE OF RUN)
 9005 FORMAT(15X,2A10)
C
      END
```

PROGRAM LISTING (Continued)

```
      SUBROUTINE INCTRL
C-----------------------------------------------------------------------
C
C * THIS ROUTINE CONTROLS INPUT READING AND INTERPRETATION OF USER
C *  RUN CONTROLS.
C
C----------------------------------------------FALCON R&D  JLW  4/20/79
C
C
C * PROGRAM CONTROL VARIABLES
C
      COMMON / CONTRL / LUIN , LUOUT , LUDATA , LURUN , LUREPT
C
      COMMON / CONTRL / IFSVRP , ONLINE , IANOPT , OPTION , IFBACH
      COMMON / CONTRL / USRNAM , PURPOS
C
C * TYPE DECLARATION
C
      INTEGER ONLINE , OPTION , USRNAM(2) ,PURPOS(6)
      LEVEL 2,LUIN,LUOUT,LUDATA,LURUN,LUREPT,IFSVRP,ONLINE,IANOPT
      LEVEL 2, OPTION,IFBACH,USRNAM,PURPOS
C
C * DETERMINE PROCESSING MODE
C
C * BATCH MODE REQUIRED
C
      IFBACH = 0
      ONLINE = 0
      CALL IAINPT
C
      END
```

```
      SUBROUTINE INFILE
C-----------------------------------------------------------------------
C
C * THIS ROUTINE PREPARES ALL NECESSARY FILES FOR THE RUN.
C
C---------------------------------------------FALCON R&D  JLW  4/20/79
C
C
C * PROGRAM CONTROL VARIABLES
C
      COMMON / CONTRL / LUIN , LUOUT , LUDATA , LURUN , LUREPT
C
      COMMON / CONTRL / IFSVRP , ONLINE , IANOPT , OPTION , IFBACH
      COMMON / CONTRL / USRNAM , PURPOS
C
C * TYPE DECLARATION
C
      INTEGER ONLINE , OPTION , USRNAM(2) ,PURPOS(6)
      LEVEL 2, LUIN,LUOUT,LUDATA,LURUN,LUREPT,IFSVRP,ONLINE,IANOPT
      LEVEL 2, OPTION,IFBACH,USRNAM,PURPOS
C
      REWIND LUDATA
C
C * RESERVE PF DEVICE FOR REPORT FILE
C
C
 9000 RETURN
      END
```

PROGRAM LISTING (Continued)

```
      SUBROUTINE KITTY8
C-------------------------------------------------------------
C
C * SUBROUTINE KITTY8 CONTROLS THE ANALYSIS OF GUN DATA,
C      BY SELECTING VARIOUS OPTIONS
C
C------------------------------------FALCON R&D  MEK 6-19-79
      COMMON / CONTRL / LUIN , LUOUT , LUDATA , LURUN , LUREPT
      COMMON / CONTRL / IFSVRP , ONLINE , IANOPT , OPTION , IFBACH
      COMMON / CONTRL / USRNAM , PURPOS
      COMMON /COMOPT/ SUBOPT
      INTEGER ONLINE , OPTION , USRNAM(2) , PURPOS(6)
      INTEGER SUBOPT
      LEVEL 2, LUIN,LUOUT,LUDATA,LURUN,LUREPT,ONLINE,IFSVRP,IANOPT
      LEVEL 2,OPTION,USRNAM,PURPOS,SUBOPT
      LEVEL 2,IFBACH
C
C * READ IN COMMON SDATA
C
      CALL OVERLAY(1HB,1,0)
 1000 CONTINUE
      READ (LUIN ,9002) SUBOPT
 3000 CONTINUE
      WRITE (6,9003) SUBOPT
      IF (SUBOPT .EQ. 0) GO TO 9000
      IF (SUBOPT .EQ. 11) GO TO 3500
      IF (SUBOPT .EQ. 12) GO TO 3750
      IF (SUBOPT .LT. 1 .OR. SUBOPT .GT. 5) CALL PERROR(79)
      GO TO (4000, 5000, 6000, 7000, 7500), SUBOPT
      GO TO 9000
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
 3500 CONTINUE
C
C * SELECT ANALYSIS OPTION DERIV -- FIRST AND
C       SECOND DERIVATIVES OF ENTIRE DATA SET.
C
      CALL OVERLAY(1HL,7,0)
      GO TO 8000
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
 3750 CONTINUE
C
C * SELECT ANALYSIS OPTION DETECT -- FIND FIRST AND
C      SECOND EPOCHS IN DATA SET.
C
      CALL OVERLAY(1HN,8,0)
      GO TO 8000
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
 4000 CONTINUE
C
C * SELECT ANALYSIS OPTION SUBSET -- CHOOSE SUBSET OF DATA.
C
      CALL OVERLAY(1HC,2,0)
      GO TO 8000
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
 5000 CONTINUE
C
C * SELECT ANALYSIS OPTION PAWS -- POWER SPECTRUM ANALYSIS
```

PROGRAM LISTING (Continued)

```
C
      CALL OVERLAY(1HD,3,0)
      GO TO 8000
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
 6000 CONTINUE
C
C * SELECT ANALYSIS OPTION GIRAFE -- FORWARD BACKWARD LEAST SQUARES FITS
C        NOT CURRENTLY AVAILABLE
C
      CALL OVERLAY(1HE,4,0)
      GO TO 8000
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
 7000 CONTINUE
C
C * SELECT ANALYSIS OPTION THEFFT -- CREATES 4TH DEGREE
C        POLYNOMIAL, PERIODIC FUNCTION OF FIRST DEGREE
C        FREQUENCIES TO FIT DATA SUBSET.
C
      CALL OVERLAY(1HF,5,0)
      GO TO 8000
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
 7500 CONTINUE
C
C * SELECT ANALYSIS OPTION SMOOTH -- 11-POINT LOCAL
C        AVERAGE OR 11-POINT LOCAL MEDIAN.
C
      CALL OVERLAY(1HG,6,0)
      GO TO 8000
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
 8000 CONTINUE
      GO TO 1000
 9000 RETURN
 9003 FORMAT(1H1,////,10X,19HANALYSIS OPTION IS ,I2,/,15X,
     *      11H1  = SUBSET,/,15X,
     *      9H2  = PAWS,/,15X,11H3  = GIRAFE,/,15X,11H4  = THEFFT,
     *      /,15X,11H5  = SMOOTH,/,15X,10H11 = DERIV,/,15X,
     *      11H12 = DETECT)
 9002 FORMAT (I2)
      END
```

PROGRAM LISTING (Continued)

```
      SUBROUTINE PERROR(IERRNO)
C-----------------------------------------------------------------------
C
C * ERROR REPORTING ROUTINE - SEE USER DIAGNOSTICS FOR MORE INFORMATION.
C
C--------------------------------------------------FALCON R&D  JLW  4/20/79
C
C
C * PROGRAM CONTROL VARIABLES
C
      COMMON / CONTRL / LUIN , LUOUT , LUDATA , LURUN , LUREPT
C
      COMMON / CONTRL / IFSVRP , ONLINE , IANOPT , OPTION , IFBACH
      COMMON / CONTRL / USRNAM , PURPOS
C
C * TYPE DECLARATION
C
      INTEGER ONLINE , OPTION , USRNAM(2) ,PURPOS(6)
      LEVEL 2, LUIN,LUOUT,LURUN,LUDATA,LUREPT,IFSVRP,ONLINE,IANOPT
      LEVEL 2, OPTION,IFBACH,USRNAM,PURPOS
C
      WRITE(LUOUT,9001) IERRNO
C
C *   SPECIAL ERROR PROCESSING GOES HERE.
C
 9000 STOP " ABNORMAL TERMINATION *** SEE OUTPUT AND USER DIAGNOSTICS**"
C
 9001 FORMAT(1X,19HPROGRAM ERROR TYPE ,I3, 19H HAS BEEN DETECTED ,/,
     1       1X,41HSEE USER DIAGNOSTICS FOR MORE INFORMATION,10(1H*))
C
      END
```

PROGRAM LISTING (Continued)

```
      SUBROUTINE RDPURP
C------------------------------------------------------------------------
C
C * THIS ROUTINE ACCEPTS USER INPUT TO DOCUMENT THE PURPOSE OF THE RUN.
C * THE TEXT IS WRITTEN TO A RUN INFO FILE TAPE2 (LURUN = 2)
C
C----------------------------------------------------FALCON R&D JLW 6/5/79
C
C
C * PROGRAM CONTROL VARIABLES
C
      COMMON / CONTRL / LUIN , LUOUT , LUDATA , LURUN , LUREPT
C
      COMMON / CONTRL / IFSVRP , ONLINE , IANOPT , OPTION , IFBACH
      COMMON / CONTRL / USRNAM , PURPOS
C
C * TYPE DECLARATION
C
      INTEGER ONLINE , OPTION , USRNAM(2) ,PURPOS(6)
      LEVEL 2, LUIN,LUOUT,LUDATA,LURUN,LUREPT,IFSVRP,ONLINE,IANOPT
      LEVEL 2, OPTION,IFBACH,USRNAM,PURPOS,ILINES,NLINES
      COMMON / REPCOM / ILINES , NLINES
C
      READ(LUIN,9001) NLINES
      DO 3000 ILINES=1,NLINES
 1000 READ(LUIN,9003)(PURPOS(I),I=1,6)
      WRITE(LUOUT,9002) (PURPOS(I),I=1,6)
      WRITE(LURUN,9002) (PURPOS(I),I=1,6)
C
 3000 CONTINUE
C
C * END OF INPUT TEXT, REWIND FILE AND EXIT.
C
 7000 REWIND LURUN
C
 9000 RETURN
C
 9001 FORMAT(I5)
C
 9002 FORMAT(15X,6A10)
 9003 FORMAT(6A10)
C
      END
```

PROGRAM LISTING (Continued)

```
      SUBROUTINE SKPDAT
      COMMON/CONTRL/LUIN,LUOUT,LUDATA,LURUN,LUREPT
      COMMON/CONTRL/IFSVRP,ONLINE,IANOPT,OPTION,IFBACH
      COMMON/CONTRL/USRNAM,PURPOS
      INTEGER ONLINE,OPTION,USRNAM(2),PURPOS(6)
      LEVEL 2,LUIN,LUOUT,LUDATA,LURUN,LUREPT,ONLINE,IFSVRP,IANOPT
      LEVEL 2,OPTION,USRNAM,PURPOS,IFBACH
C
C     READ DATA ONLY - DON'T PROCESS
C     ILOOP= NO. OF CHANNELS TO BE SKIPPED
C
      READ(LUIN,9001)ILOOP
      DO 1000 IL=1,ILOOP
      CALL OVERLAY(1HB,1,0)
 1000 CONTINUE
      READ(LUIN,9002)IANOPT
 9000 RETURN
 9001 FORMAT(I5)
 9002 FORMAT(I2)
      END
```

PROGRAM LISTING (Continued)

```
      SUBROUTINE START
C-----------------------------------------------------------------------
C
C * THIS ROUTINE HANDLES START UP PROCESSING.
C
C-----------------------------------------------FALCON R&D  JLW  4/20/79
C
C
C * PREPARE FILES FOR THE RUN.
C
      CALL INFILE
C
C * READ AND INTERPRET USER CONTROLS FOR THIS RUN
C
      CALL INCTRL
C
C * OTHER REQUIRED INITIALIZATION FOR THIS RUN.
C
      RETURN
      END
```

PROGRAM LISTING (Continued)

```
      OVERLAY(B,1,0)
      PROGRAM DATAIN
C-----------------------------------------------------------------------
C
C * SUBROUTINE DATAIN READS MUZZLEDEVICE DATA FROM TAPE1
C
C--------------------------------------------------FALCON R&D  NEA 6-7-79
      COMMON / SDATA / NSTPTS , DTIME , STRING(5000) , STIME(5000)
      COMMON / LABEL / KTB(24) , TSTART , I
      COMMON / REPDAT / REPSTR(5000) , IFREP , NDREP
      LEVEL 2, REPSTR, IFREP, NDREP
      LEVEL 2, NSTPTS, DTIME, STRING, STIME, KTB, I, TSTART
C
C * READ AND WRITE LABEL
C
      READ (1) KTB
      WRITE (6, 9001) KTB
C
C * READ AND WRITE NSTPTS = NUMBER OF DATA POINTS
C                  TSTART  = START TIME OF ARRAY
C                  DTIME   = TIME INCREMENT OF ARRAY
C
      READ (1) NSTPTS
      WRITE (6, 9002) NSTPTS
      READ (1) TSTART, DTIME
      WRITE (6, 9003) TSTART, DTIME
C
C * READ DATA IN ARRAY STRING
C
      READ (1) (STRING(I), I=1,NSTPTS)
C
C * SET UP TIME ARRAY STIME
C
      STIME(1) = TSTART
      DO 1000 I=2,NSTPTS
      STIME(I) = STIME(I - 1) + DTIME
 1000 CONTINUE
C
C * PLACE DATA STRING INTO ARRAY REPSTR.  IFREP INDICATES THE
C       REGIMES REPLACED BY THEFFT INTO REPSTR.
C
      IFREP = 0
      DO 2000 I = 1,NSTPTS
        REPSTR(I) = STRING(I)
 2000 CONTINUE
C
      CALL CLASIF
C
 9000 RETURN
 9001 FORMAT(1H1,///,20X,54(1H*),/,20X,1H*,52X,1H*,/,20X,1H*,
     *       14X,24HMUZZLEDEVICE DATA RECORD,14X,1H*,/,20X,
     *       1H*,52X,1H*,/,20X,1H*,2X,24I2,2X,1H*,/,20X,1H*,
     *       52X,1H*,/,20X,54(1H*),//)
 9002 FORMAT (10X, 21HNUMBER OF DATA POINTS, I10,/)
 9003 FORMAT (10X, 12HINITIAL TIME, F10.4, 5X, 14HTIME INCREMENT, F10.4,
     *        10X, 22H(TIME IN MILLISECONDS))
      END
```

PROGRAM LISTING (Continued)

```
      SUBROUTINE CLASIF
C-----------------------------------------------------------------------
C
C * SUBROUTINE CLASIF CLASSIFIES THE DATA SET
C       ACCORDING TO THE LEVEL OF SATURATION.
C
C-----------------------------------------------------------------------
      LEVEL 2, NLIMBL,NBLACK,NENDBL,QSTAR,NSTPTS,STRING,NBEGBL,ICLASS,I
      LEVEL 2,DTIME,STIME
      COMMON / SDATA / NSTPTS , DTIME , STRING(5000) , STIME(5000)
      COMMON / CMCLAS / NLIMBL, NBLACK, NENDBL, QSTAR, I
      COMMON / CLASCM / ICLASS, NBEGBL
C
C * DETERMINE MAXIMUM VALUE OF DATA RECORD      =QSTAR
C               FIRST OCCURRENCE                =NBEGBL
C               NUMBER OF OCCURRENCES           =NBLACK
C
C   NBLACK = 0    NO SATURATION
C                 ICLASS = 0
C
C   NBLACK < 10   LIMITED SATURATION
C                 ICLASS =1
C
C   NBLACK > 10   HEAVY SATURATION
C                 ICLASS = -1
C
      NLIMBL = 10
      NBLACK = 0
      NENDBL = 0
      NBEGBL = 0
      QSTAR = -99999.
      DO 5000 I = 1,NSTPTS
        IF (STRING(I) - QSTAR) 1000, 2000, 3000
 1000   CONTINUE
        GO TO 4000
 2000   CONTINUE
          NENDBL = I
        GO TO 4000
 3000   CONTINUE
          QSTAR = STRING(I)
          NBEGBL = I
 4000   CONTINUE
 5000 CONTINUE
      NBLACK = NENDBL - NBEGBL
      ICLASS = 0
      IF (NBLACK .LE. 0) NBLACK = 0
      IF (NBLACK .LE. 0) GO TO 6000
        ICLASS = -1
        IF (NBLACK .LT. NLIMBL) ICLASS = 1
 6000 CONTINUE
      WRITE (6,9001) ICLASS, NBLACK
      IF (ICLASS .EQ. 0) WRITE(6,9002)
      IF (ICLASS .EQ. 1) WRITE(6,9003)
      IF (ICLASS .EQ.-1) WRITE(6,9004)
 9000 RETURN
 9001 FORMAT(1X,/////,20X,27HTIME HISTORY CLASSIFICATION,//,
     *        10X,24HCLASSIFICATION NUMBER IS,I5,/,
```

PROGRAM LISTING (Continued)

```
     *        10X,35HNUMBER OF POINTS IN SIGNAL BLACKOUT,I10)
9002 FORMAT(1X,/,10X,29HCOMPLETE ANALYSIS OF DATA SET,
     *        13H IS PERMITTED,//)
9003 FORMAT(1X,/,10X,31HLIMITED SATURATION OF DATA SET,,
     *        31H COMPLETE ANALYSIS IS PERMITTED,//)
9004 FORMAT(1X,/,10X,29HHEAVY SATURATION OF DATA SET,,
     *        55H ONLY EPOCH 1 DETECTION AND REGIME I ANALYSIS PERMITTED,
     *        //)
      END
```

PROGRAM LISTING (Continued)

```
      PROGRAM SUBSET
C---------------------------------------------------------------------------
C
C * SUBROUTINE SUBSET TAKES A SUBSET OF THE DATA ARRAY STRING FOR
C      FURTHER ANALYSIS
C
C----------------------------------------FALCON R&D  NEA 6-5-79
      COMMON / SDATA / NSTPTS , DTIME , STRING(5000) , STIME(5000)
      COMMON / SSDATA / NSSPTS , SDTIME , SUBSTR(5000) , SSTIME(5000)
      LEVEL 2, NSTPTS, DTIME, STRING, STIME, NSSPTS, SDTIME,
     *         SUBSTR, SSTIME
      LEVEL 2, TINIT,TLAST,NODTS,IS,IE,IND,I,LIMP
      COMMON / COMSET / IS,IE,IND,I,LIMP
      COMMON / COMSUB / TINIT , TLAST , NODTS
C
C * READ PARAMETERS
C         TINIT = INITIAL TIME DESIRED FOR SUB-STRING
C         TLAST = FINAL TIME DESIRED FOR SUB-STRING
C         NODTS = NUMBER OF DTIME INTERVALS BETWEEN SUB-STRING POINTS
C
      CALL SUBPAR (TINIT, TLAST, NODTS)
C
C * INITIAL AND FINAL INDICES IN STRING
C
      IS = (TINIT - STIME(1))/DTIME + 1
      IE = (TLAST - STIME(1))/DTIME + 1
C
C * CHECK INDICES OF STRING ARRAY TO MATCH TIMES
C
      IF (IS .LE. 0) CALL PERROR(53)
      IF (IE .LE. IS) CALL PERROR(54)
      IF (ABS(STIME(IS) - TINIT) .GT. DTIME) CALL PERROR(60)
      IF (ABS(STIME(IE) - TLAST) .GT. DTIME) CALL PERROR(61)
      IF (TINIT .EQ. STIME(IS+1)) IS = IS + 1
      IF (TLAST .EQ. STIME(IE+1)) IE = IE + 1
C
C * CREATE SUBSTR AND SSTIME ARRAYS FROM STRING AND STIME
C
      I = 0
      DO 1000 IND=IS,IE,NODTS
        I = I + 1
        SUBSTR(I) = STRING(IND)
        SSTIME(I) = STIME(IND)
 1000 CONTINUE
C
C * SET NSSPTS TO NUMBER OF POINTS IN SUBSTR AND SSTIME
C
      NSSPTS = I
      IF (NSSPTS .LE. 0) CALL PERROR(55)
C
C * SET SDTIME TO TIME INTERVAL IN SUBSTR AND SSTIME
C
      SDTIME = DTIME*FLOAT(NODTS)
C
 9000 RETURN
      END
```

PROGRAM LISTING (Continued)

```
      SUBROUTINE SUBPAR (TINIT, TLAST, NODTS)
C-----------------------------------------------------------------------
C
C * SUBROUTINE SUBPAR READS INITIAL AND FINAL TIMES WITH POINT INTERVAL
C      FOR CHOOSING SUBSET OF SAMPLE DATA
C
C-------------------------------------------------FALCON R&D  NEA 6-5-79
      COMMON / CONTRL / LUIN , LUOUT , LUDATA , LURUN , LUREPT
      COMMON / CONTRL / IFSVRP , ONLINE , IANOPT , OPTION , IFBACH
      COMMON / CONTRL / USRNAM , PURPOS
      INTEGER ONLINE , OPTION , USRNAM(2) , PURPOS(6)
      LEVEL 2, LUIN,LUOUT,LUDATA,LURUN,LUREPT,ONLINE,IFSVRP,IANOPT
      LEVEL 2, OPTION,USRNAM,PURPOS,TINIT,TLAST,NODTS
      LEVEL 2,IFBACH
C
C * READ TINIT = INITIAL TIME
C        TLAST = FINAL TIME
C        NODTS = NUMBER OF DTIME'S BETWEEN SUBSTRING POINTS
C
C
C   IANOPT = 2 ANALYSIS OPTION IS COMPLT
C              TINIT, TLAST,NODTS ARE SPECIFIED PRIOR TO SR CALL
C
      IF (IANOPT .EQ. 2) GO TO 1000
      READ (LUIN, 9001) TINIT, TLAST, NODTS
 1000 CONTINUE
      IF (TLAST .LE. TINIT) CALL PERROR(51)
      IF (NODTS .LE. 0) CALL PERROR(52)
      IF (IANOPT .EQ. 2) GO TO 2000
      WRITE (LUOUT, 9002) TINIT, TLAST, NODTS
 2000 CONTINUE
 9000 RETURN
 9001 FORMAT (2F10.4, I10)
 9002 FORMAT(1X,/,28H DATA SUBSET BEGINS AT TIME ,F10.4,
     *        28H MILLISECONDS, ENDS AT TIME , F10.4,
     *        37H MILLISECONDS, AND TAKES POINTS EVERY, I5, 7H DTIMES,/)
      END
```

PROGRAM LISTING (Continued)

```
      OVERLAY(D,3,0)
      PROGRAM PAWS
C-----------------------------------------------------------------------
C
C * SUBROUTINES PAWS CALCULATES THE POWER SPECTRUM FOR ARRAY SUBSTR
C     USING SYSTEM SUBROUTINE FTFREQ
C     CALLED BY COMPLT (IANOPT=2) OR KITTY8 (IANOPT=1)
C     PAWS = POWER ANALYSIS OF WAVEFORM SPECTRA
C
C------------------------------------------FALCON R&D  NEA 6-5-7
      COMMON / CONTRL / LUIN , LUOUT , LUDATA , LURUN , LUREPT,
     *                  IFSVRP , ONLINE , IANOPT , OPTION , IFBACH,
     *                  USRNAM , PURPOS
      COMMON / SSDATA / NSSPTS , SDTIME , SUBSTR(5000) , SSTIME(5000)
      COMMON / PAWCOM / XYMV(6), ACV(1602), FREQ(801), PS(1602),
     *                  XCOV(1602), XSPECT(1602), AMPHAS(1602),
     *                  XFER(1602), COHER(801)
      COMMON /FREQPS/ A(810) , B(810) , R(1620) , NROW
      COMMON /FREQS/ FREQEN(20) , NFR
      COMMON / SCRIB1 / XARR(100) , YARR(100) , DX1 , NDPTS , NPP1
      COMMON / LAGCOM / M
      COMMON / PAWVAR / X(1500) , IND(6) , XIND(2) , PLOTOP , LAGOPT,
     *                  I, NFREQ
C
      INTEGER ONLINE , OPTION , USRNAM(2) , PURPOS(6) , PLOTOP
C
      LEVEL 2, NSSPTS, SDTIME, SUBSTR, SSTIME, A, B, R, FREQEN, NFR
     *       , LUIN,LUOUT,LUDATA,LURUN,LUREPT,IFSVRP,ONLINE,IANOPT
     *       , OPTION,USRNAM,PURPOS,M,NROW
     *       , IFBACH,XYMV,ACV,FREQ,PS,XCOV,XSPECT,AMPHAS,XFER,COHER
     *       , X,IND,XIND,PLOTOP , LAGOPT,I,NFREQ
C
C * WRITE PAGE HEADING FOR PAWS
C
      WRITE(6,9005)
C
C   PLOTOP = PLOTTING OPTION FOR POWER SPECTRUM
C          = 0  OFF
C          = 1  ON
C
C   LAGOPT = FREQUENCY MULTIPLE OPTION FOR POWER SPECTRUM
C          = 0  MINIMUM ALLOWABLE FREQUENCY MULTIPLE (.25,.5,1 OR 2)
C          = 1  FREQUENCY MULTIPLE IS 1.0
C
      READ(5,9003)  PLOTOP , LAGOPT
      WRITE(6,9004) PLOTOP , LAGOPT
C
C * PARAMETERS
C        M = NUMBER OF LAG TIMES
C
      M = 2. * (1. / SDTIME)
      IF (LAGOPT .EQ. 1) GO TO 500
      IF (M .GT. NSSPTS) M = M / 2
      IF (M .GT. NSSPTS) M = M / 2
      IF (M .GT. NSSPTS) M = M / 2
      GO TO 750
  500 CONTINUE
```

PROGRAM LISTING (Continued)

```
      M = M / 4
      IF (M .GT. NSSPTS) M = M / 2
      IF (M .GT. NSSPTS) M = M / 2
C
C * INITIALIZE FTFREQ ARRAYS
C
  750 CONTINUE
      DO 1000 I=1,NSSPTS
        X(I) = SUBSTR(I)
 1000 CONTINUE
      IND(1)  = 0
      IND(2)  = NSSPTS
      IND(3)  = 0
      IND(4)  = M
      IND(5)  = 0
      IND(6)  = 0
      XIND(1) = SDTIME
      XIND(2) = 0
C
C * CHECK INPUT TO SUBROUTINE FTFREQ
C
      IF (IND(2) .LE. 3) CALL PERROR(57)
      IF (IND(4) .LE. 2) CALL PERROR(58)
      IF (IND(4) .GE. IND(2)) CALL PERROR(59)
C
C * DO SPECTRUM ANALYSIS
C
      CALL FTFREQ (X, IND, XIND, XYMV, ACV, FREQ, PS,
     *             XCOV, XSPECT, AMPHAS, XFER, COHER, IER)
      IF (IER .NE. 0) WRITE (LUOUT, 9001) IER
C
C * PRINT OUTPUT FROM POWER SPECTRUM
C
      CALL PAWOUT
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C * PUT TOP 10 MAJOR FREQUENCIES (FREQUENCIES WITH LARGEST
C        POWER SPECTRUM) INTO FREQEN.
C
      NFR = 10
      NROW = M + 1
      DO 3000 I = 1,NROW
        A(I) = FREQ(I)
        B(I) = PS(I)
 3000 CONTINUE
C
C * SORT FREQUENCIES INTO R BY DECREASING POWER
C
      CALL ROWSRT(A,B,R,NROW)
C
C * PUT 10 LARGEST (FIRST 10) INTO FREQEN
C
      DO 4000 I = 1,NFR
        FREQEN(I) = R(I)
 4000 CONTINUE
C
C * WRITE 10 PREDOMINANT FREQUENCIES
```

PROGRAM LISTING (Continued)

```
C
      WRITE(6,9002) (FREQEN(I),I=1,NFR)
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      IF (PLOTOP .EQ. 0) GO TO 7000
C
C * PLOT LOG(POWER) VS FREQUENCIES (FIRST 25)
C
      NFREQ = M + 1
      DO 5000 I = 1,NFREQ
        IF (PS(I) .LT. 1) PS(I) = 1.
        PS(I) = ALOG10(PS(I))
 5000 CONTINUE
      DX1 = FREQ(2) - FREQ(1)
      NDPTS = MIN0(NFREQ,IFIX(10. / DX1 + 1.0))
      NDPTS=MAX0(NDPTS,20)
      DO 6000 I = 1,NDPTS
        XARR(I) = FREQ(I)
        YARR(I) = PS(I)
 6000 CONTINUE
      NPPI = IFIX(1.0 / DX1)
      NPPI=MAX0(NPPI,2)
      CALL OVERLAY(1HK,3,1)
 7000 CONTINUE
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
 9000 RETURN
 9001 FORMAT(1X,/,10X,46H**WARNING** ERROR TERM FROM SUBROUTINE FTFREQ ,
     A        26H(DEFINED BY SUBROUTINE) IS, I5, //)
 9002 FORMAT (1X,///,20X,27HTEN PREDOMINANT FREQUENCIES,
     A       //,5X,10F10.5,/)
 9003 FORMAT(2I2)
 9004 FORMAT(1X,////,10X,23HPLOT OPTION FOR PAWS IS,5X,I5,2X,'LAGOPT = '
     *       ,I5)
 9005 FORMAT(1H1,///,20X,54(1H*),/,20X,1H*,52X,1H*,/,20X,1H*,15X,
     *       23HPOWER SPECTRUM ANALYSIS,14X,1H*,/,20X,1H*,52X,
     *       1H*,/,20X,54(1H*))
      END
```

PROGRAM LISTING (Continued)

```
      SUBROUTINE PAWOUT
C--------------------------------------------------------------------
C
C * SUBROUTINE PAWOUT PRINTS THE OUTPUT FROM SUBROUTINE PAWS
C
C--------------------------------------------------FALCON R&D  NEA 6-5-79
      COMMON / CONTRL / LUIN , LUOUT , LUDATA , LURUN , LUREPT
      COMMON / CONTRL / IFSVRP , ONLINE , IANOPT , OPTION , IFBACH
      COMMON / CONTRL / USRNAM , PURPOS
      INTEGER ONLINE , OPTION , USRNAM(2) , PURPOS(6)
      COMMON / PAWCOM / XYMV(6), ACV(1602), FREQ(801), PS(1602),
     *                  XCOV(1602), XSPECT(1602), AMPHAS(1602),
     *                  XFER(1602), COHER(801)
      LEVEL 2, LUIN,LUOUT,LUDATA,LURUN,LUREPT,IFSVRP,ONLINE,IANOPT
     *       , OPTION,USRNAM,PURPOS,M,LAG,FIND
     *       , XYMV,ACV,FREQ,PS,XCOV,XSPECT,AMPHAS,XFER,COHER
      LEVEL 2,IFBACH
      COMMON /PCOMM/ LAG , FIND
      COMMON / LAGCOM / M
C
C * PRINT HEADING, MEAN, VARIANCE
C
      WRITE (LUOUT, 9001)
      WRITE (LUOUT, 9002) XYMV(1), XYMV(2)
C
C * PRINT LAG AND AUTOCOVARIANCE (LAG) TABLE
C         FREQUENCY AND POWER SPECTRUM TABLE
C
      WRITE (LUOUT, 9003)
      LAG = 0
      WRITE (LUOUT, 9004) LAG, XYMV(2), FREQ(1), PS(1)
      DO 1000 LAG=1,M
        FIND = LAG + 1
        WRITE (LUOUT, 9004) LAG, ACV(LAG), FREQ(FIND), PS(FIND)
 1000 CONTINUE
      WRITE (LUOUT, 9005)
 9000 RETURN
 9001 FORMAT (1H1, //, 20X, 34HRESULTS OF POWER SPECTRUM ANALYSIS, // )
 9002 FORMAT (1X, 10X, 13HWAVEFORM MEAN, F15.4, /, 11X,
     *        17HWAVEFORM VARIANCE, F15.4, /)
 9003 FORMAT (1X, //, 17X, 20HAUTOCOVARIANCE TABLE, 10X,
     *        14HPOWER SPECTRUM, //, 17X, 3HLAG, 3X, 14HAUTOCOVARIANCE,
     *        5X, 9HFREQUENCY, 3X, 14HPOWER SPECTRUM, /)
 9004 FORMAT (10X, I10, F15.5, 5X, F10.5, F15.5)
 9005 FORMAT (1X, ///, 25X, 30HEND OF POWER SPECTRUM ANALYSIS, ////)
      END
```

PROGRAM LISTING (Continued)

```
      SUBROUTINE ROWSRT(A,B,R,N)
C-----------------------------------------------------------------------
C
C * SUBROUTINE ROWSRT SORTS ROWS OF A MATRIX A ACCORDING TO DESCENDING
C     ORDER OF MATRIX B.  STORE SORTED A INTO R.
C
C-----------------------------------------------------------------------
      DIMENSION A(810), B(810), R(1620)
      COMMON /SRTCOM/ I,I2,ISORT,RSAVE,SAVER,IR,IA
      LEVEL 2, A,B,R,I,N,I2,ISORT,RSAVE,SAVER,IR,IA
C
C * MOVE SORTING KEY VECTOR TO FIRST COLUMN OF OUTPUT MATRIX AND
C     BUILD ORIGINAL SEQUENCE LIST IN SECOND COLUMN
C
      DO 1000 I = 1,N
        R(I) = B(I)
        I2 = I + N
        R(I2) = I
 1000 CONTINUE
C
C * SORT ELEMENTS IN SORTING KEY VECTOR (ORIGINAL SEQUENCE LIST IS
C     RESEQUENCED ACCORDINGLY)
C
 2000 CONTINUE
      ISORT = 0
      DO 4000 I = 2,N
        IF (R(I) - R(I-1)) 4000, 4000, 3000
 3000   CONTINUE
        ISORT = ISORT + 1
        RSAVE = R(I)
        R(I) = R(I-1)
        R(I-1) = RSAVE
        I2 = I + N
        SAVER = R(I2)
        R(I2) = R(I2-1)
        R(I2-1) = SAVER
 4000 CONTINUE
      IF (ISORT) 2000, 5000, 2000
C
C * MOVE ROWS FROM MATRIX A TO MATRIX R (NUMBER IN SECOND COLUMN OF R
C     REPRESENTS ROW NUMBER OF MATRIX A TO BE MOVED)
C
 5000 CONTINUE
      DO 8000 I = 1,N
C
C * MOVE ELEMENT TO OUTPUT MATRIX
C
        I2 = I + N
        R(I) = A( R(I2) )
 8000 CONTINUE
      RETURN
      END
```

PROGRAM LISTING (Continued)

```
      OVERLAY(K,3,1)
      PROGRAM SCRIB1
C-----------------------------------------------------------------
C
C * SCRIB1 PLOTS GRAPHS FOR MUZZLEDEVICE DATA IN PAWS
C       USING CALCOMP PLOTTING PACKAGE
C
C-----------------------------------------------------------------
      COMMON / SCRIB1 / XARRAY(100) , YARRAY(100) , DX1 , NDPTS , NPPI
      REAL        DX , XSCALE , YSCALE ,XLNGTH .
      REAL        YLNGTH , XOR , YOR , XBASE , YBASE , FACTOR
      INTEGER     ISENT(10) , LABEL(4) , IUNIT
      COMMON / LABEL / KTB(24) , TSTART , I
      LEVEL 2, KTB, TSTART, I
C
      DATA LABEL/'BOOTS','B390','X3983','M392 DATA'/
C
C                TEST ARGUMENTS FOR ERRORS
C
      IF (NDPTS .GT. 100) CALL PERROR (82)
C
C                DEFINE VARIABLES USED BY PLOTTING ROUTINES
C
      XLNGTH=11.
      YLNGTH=8.5
      XSIZE=8.0
      YSIZE=6.0
      XBASE=2.0
      YBASE=1.0
      CALL FIXSCA(XARRAY(1),NDPTS,XSIZE,XSCALE,XOR,XMAX,DX)
      CALL FIXSCA(YARRAY(1),NDPTS,YSIZE,YSCALE,YOR,YMAX,DY)
      XLNGTH=11.
      YLNGTH=8.5
      XSIZE=8.0
      YSIZE=6.0
      XBASE=2.0
      YBASE=1.0
      CALL FIXSCA(XARRAY(1),NDPTS,XSIZE,XSCALE,XOR,XMAX,DX)
      CALL FIXSCA(YARRAY(1),NDPTS,YSIZE,YSCALE,YOR,YMAX,DY)
      FACTOR = 1.
      IUNIT  = 10
C
C                BRL PLOTTING ROUTINES (INITIAL)
C
      CALL PLTBEG(XLNGTH,YLNGTH,FACTOR,IUNIT,LABEL)
      CALL PLTSCA(XBASE,YBASE,XOR,YOR,XSCALE,YSCALE)
C
C                PLOT THE CURVES
C
        CALL PLTDTS(1,0,XARRAY,YARRAY,NDPTS,0)
C
C                 PLOT AXES AND LABELS
C
      MODE = 4
      ONE = 1.
      CALL PLTAXS(DX,DY,XOR,XMAX,YOR,YMAX,MODE)
      CALL LABELA(DX,DY,XOR,XMAX,YOR,YMAX,ONE,ONE)
```

PROGRAM LISTING (Continued)

```
C
      YMAX1=YMAX+1.0*YSCALE
      YMAX1=YMAX+1.0*YSCALE
      ENCODE (100, 130, ISENT)  KTB
  130 FORMAT (1X, 24I2, '>')
      CALL PLTSYM(0.15,ISENT,0.,XOR,YMAX1)
      CALL PLTSYM(0.15,ISENT,0.,XOR,YMAX1)
C
      CALL PLTPGE
      RETURN
      END
```

PROGRAM LISTING (Continued)

```
      OVERLAY(E,4,0)
      PROGRAM GIRAFE
C----------------------------------------------------------------------
C
C * SUBROUTINE GIRAFE DOES A FORWARD AND BACKWARD INTERPOLATION
C      FROM EACH POINT IN THE STRING AND CALCULATES THE
C      ERROR FOR THE EXTRAPOLATED POINT
C
C-------------------------------------------FALCON R&D   6/28/79  NEA
$     COMMON / SSDATA / NSSPTS, SDTIME, SUBSTR(5000), SSTIME(5000)
$     DIMENSION ERR(20,2), WKAREA(50), A(10,4), B(10,1), C(10)
$     LEVEL 2, NSSPTS, SDTIME, SUBSTR, SSTIME
$     LEVEL 2, NLIM,IST,LLST,INDEX,IND,XI,IWK,C,YI,YF,YB,ERR
$     COMMON / CMGIRF / NLIM,IST,LLST,INDEX,IND,XI,IWK,C,YI,YF,YB,ERR
C
C * READ NUMBER OF DATA POINTS FOR INTERPOLATION FIT
C
$     READ (5, 9004) NBATS
C
C * CALCULATE NUMBER OF EXTRAPOLATION POINTS
C
$     NLIM = NSSPTS - 2*NBATS
$     NCOEF = 4
$     NDIM = 10
$     NONE = 1
$     NDGT = 7
C
C * LOOP FOR FORWARD, BACKWARD INTERPOLATION
C      INTERMEDIATE EXTRAPOLATION
C
$     DO 1000 IST=1,NLIM
C
C * FORWARD EXTRAPOLATION
C
$     LLST = IST
$     DO 2000 INDEX = 1,NBATS
$       IND = LLST + INDEX - 1
$       XI = SSTIME(IND)
$       A(INDEX,1) = XI**3
$       A(INDEX,2) = XI**2
$       A(INDEX,3) = XI
$       A(INDEX,4) = 1.0
$       B(INDEX,1) = SUBSTR(IND)
$2000 CONTINUE
C
C * LEAST SQUARES FIT
C
$     DO 2500 IWK=1,50
$     WKAREA(IWK) = 0.0
$2500 CONTINUE
$     CALL LLSQAR (A, B, NBATS, NCOEF, NONE, NDIM, NDIM, NDGT, WKAREA,
$    *             IER)
$     DO 3000 IND=1,4
$       C(IND) = B(IND,1)
$3000 CONTINUE
$     XI = SSTIME(IST + NBATS)
$     YI = SUBSTR(IST + NBATS)
```

PROGRAM LISTING (Continued)

```
$      YF = C(1)*XI**3 + C(2)*XI**2 + C(3)*XI + C(4)
$      WRITE (6, 9001) (C(I), I=1,4)
$      DO 3500 INDEX = 1,NBATS
$        IND = LLST + INDEX - 1
$        XI = SSTIME(IND)
$        YI = C(1)*XI**3 + C(2)*XI**2 + C(3)*XI + C(4)
$        WRITE(6,9006) XI , SUBSTR(IND) , YI
$3500 CONTINUE
$      WRITE (6, 9007) XI, YI, YF
C
C * BACKWARD EXTRAPOLATION
C
$      LLST = IST + NBATS + 1
$      DO 4000 INDEX = 1,NBATS
$        IND = LLST + INDEX - 1
$        XI = SSTIME(IND)
$        A(INDEX,1) = XI**3
$        A(INDEX,2) = XI**2
$        A(INDEX,3) = XI
$        A(INDEX,4) = 1.0
$        B(INDEX,1) = SUBSTR(IND)
$4000 CONTINUE
C
C * LEAST SQUARES FIT
C
$      DO 4500 IWK=1,50
$      WKAREA(IWK) = 0.0
$4500 CONTINUE
$      CALL LLSQAR (A, B, NBATS, NCOEF, NONE, NDIM, NDIM, NDGT, WKAREA,
$    *                 IER)
$      DO 5000 IND=1,4
$        C(IND) = B(IND,1)
$5000 CONTINUE
$      XI = SSTIME(IST + NBATS)
$      YI = SUBSTR(IST + NBATS)
$      YB = C(1)*XI**3 + C(2)*XI**2 + C(3)*XI + C(4)
$      WRITE (6, 9002) (C(I), I=1,4)
$      WRITE (6, 9007) XI, YI, YB
$      DO 5500 INDEX = 1,NBATS
$        IND = LLST + INDEX - 1
$        XI = SSTIME(IND)
$        YI = C(1)*XI**3 + C(2)*XI**2 + C(3)*XI + C(4)
$        WRITE(6,9006) XI , SUBSTR(IND) , YI
$5500 CONTINUE
C
C * CALCULATE ERROR
C
$      ERR(IST,1) = SSTIME(IST + NBATS)
$      ERR(IST,2) = YF - YB
$      WRITE (6, 9003) (ERR(IST,I), I=1,2)
$1000 CONTINUE
 9000 RETURN
$9001 FORMAT (1X, //, 21HFORWARD EXTRAPOLATION, /,
$    *           26HLEAST SQUARES COEFFICIENTS, 4E14.7)
$9002 FORMAT (1X, /, 22HBACKWARD EXTRAPOLATION, / ,
$    *           26HLEAST SQUARES COEFFICIENTS, 4E14.7)
$9003 FORMAT (1X, /, 4HTIME, E14.7, 5X, 22HFORWARD-BACKWARD ERROR,
```

PROGRAM LISTING (Continued)

```
$     *          E14.7,///)
$9004 FORMAT  (I2)
$9005 FORMAT  (1X, 37HLST SQR DATA: INDEX, X3, X2, X, 1, Y ,I5,5E14.7)
$9006 FORMAT(1X,4HTIME,2X,E14.7,2X,5HVALUE,2X,E14.7,2X,
$     *          8HESTIMATE,2X,E14.7)
$9007 FORMAT  (1X,/, 4HTIME,5X,E14.7,10X,5HVALUE,5X,E14.7,10X,
$     *          8HESTIMATE,5X,E14.7,/)
      END
```

PROGRAM LISTING (Continued)

```
      OVERLAY(F,5,0)
      PROGRAM THEFFT
C----------------------------------------------------------------------
C
C * SUBROUTINE THEFFT ANALYSIS IS FOR THE FIRST OR THIRD REGIME
C     POLYNOMIAL FIT, FAST FOURIER TRANSFORM
C     REMAINDER AFTER SUBTRACTION OF FITS IS NOISE
C
C----------------------------------------------------------------------
      COMMON / CONTRL / LUIN , LUOUT , LUDATA , LURUN , LUREPT
      COMMON / CONTRL / IFSVRP , ONLINE , IANOPT , OPTION , IFBACH
      COMMON / CONTRL / USRNAM , PURPOS
      COMMON / SSDATA / NSSPTS, SDTIME, SUBSTR(5000), SSTIME(5000)
      COMMON / SDATA  / NSTPTS , DTIME , STRING(5000) , STIME(5000)
      COMMON / REPDAT / REPSTR(5000) , IFREP ,  NDREP
      COMMON / FREQS   / FREQEN(20), NFR
      COMMON / CMLOOT / AC(20), BS(20), INDEX(1500),
     *                  CSTR(1500), DSTR(1500), PER(1500),
     *                  RNOISE(1500),  I, A0, OMEGA, PI,
     *                  J, K, SUMER, AVEER, SIGMA, PLOTOP(2),
     *                  NBIN, CELLS(10), COMP(10), CS, IDF, Q,
     *                  IIND,SUMD, AVED, SIGD, SIGNDD, STAR,
     *                  TIME, IREP ,ISTART, ISTOP, I1, I2
      COMMON / SCRIB  / XARR(5000) , YARR1(5000) , YARR2(5000),
     A                  YARR3(5000) , NARR , DELTAX , NPOINT , NPPI
      COMMON / POLYCM / BSTR(1500), MAXDEG, POLY(1500), COEF(10), NCOEF
      COMMON / DERIVS / STRDOT(5000), STRDD(5000), PLTOP1
C
      LEVEL 2, NSSPTS, SDTIME, SUBSTR, SSTIME, FREQEN, NFR,
     *         AC, BS, MAXDEG, COEF, BSTR, CSTR, DSTR, PER,
     *         RNOISE, POLY, I, A0, OMEGA, PI, J, K, SUMER,
     *         AVEER, SIGMA, NCOEF, PLOTOP,  INDEX,
     *         NBIN, CELLS, COMP, CS, IDF, Q, NDREP,
     *         NSTPTS,DTIME,STRING,STIME,REPSTR,IFREP,
     *         STRDOT, STRDD, PLTOP1, IIND, SUMD, AVED, SIGD,
     *         SIGNDD, STAR,TIME,IREP,ISTART,ISTOP,I1,I2
      LEVEL 2,LUIN,LUOUT,LUDATA,LURUN,LUREPT,IFSVRP,ONLINE
      LEVEL 2,IANOPT,OPTION,IFBACH,USRNAM,PURPOS
C
      INTEGER PLOTOP , PLTOP1
      EXTERNAL NORM
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C * SET UP CONSTANTS
C
      NDMAX = 1500
      NDPTS = NSSPTS
      PI = 3.1415926
      NCOEF = 4
      NBIN = 8
      IDF = 0
      MAXDEG = 4
      IF (NSSPTS .GT. NDMAX) CALL PERROR (81)
C
C * WRITE PAGE HEADING FOR THEFFT
C
```

PROGRAM LISTING (Continued)

```
      WRITE(6,9012)
C
C * PLOTOP(2) = PLOT OPTIONS FOR DATA SUBSET ANALYSIS
C               = 1   ON
C               = 0   OFF
C         (1) = 3 PLOTS OF DATA, POLYNOMIAL FIT, PERIODIC FIT
C                 AND NOISE
C         (2) = 1 PLOT OF DATA SET AFTER NOISE HAS BEEN REMOVED
C
      READ(5,9002)  PLOTOP
      WRITE(6,9009) PLOTOP
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C
C * BSTR IS ORIGINAL STRING
C
      DO 1000 I = 1,NDPTS
        BSTR(I) = SUBSTR(I)
 1000 CONTINUE
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C * FOURTH DEGREE ORTHOGONAL POLYNOMIAL FIT ON BSTR1 IN POLY
C     CSTR IS ORIGINAL MINUS POLYNOMIAL, WHICH IS THE LEVELED
C     CURVE OF THE ORIGINAL.
C
      CALL OVERLAY(1HH,5,1)
      WRITE(6,9011) (COEF(I) , I = 1,NCOEF)
      DO 2000 I = 1,NDPTS
        CSTR(I) = BSTR(I) - POLY(I)
        SUBSTR(I) = CSTR(I)
 2000 CONTINUE
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C * OBTAIN 10 MAJOR FREQENCIES OF CSTR IN FREQEN BY
C        SR PAWS - POWER SPECTRUM.
C
      CALL OVERLAY(1HI,5,2)
C
C * CALCULATE COEFFICIENTS OF COSINES, SINES OF MAJOR FREQUENCIES
C
      A0 = 0.
C
C * CONSTANT A0
C
      DO 2750 I = 1,NFR
        IF (FREQEN(I) .NE. 0) GO TO 2750
          A0 = 0.
          GO TO 3500
 2750 CONTINUE
      DO 3000 J = 1,NDPTS
        A0 = A0 + 2. * CSTR(J)
 3000 CONTINUE
 3500 CONTINUE
C
C * COEFFICIENTS AC(K) AND BS(K) FOR COSINE AND SINE OF
```

PROGRAM LISTING (Continued)

```
C       FREQUENCY FREQEN(K)
C
      WRITE(6,9019)
      WRITE(6,9013)
      DO 4000 I = 1,NFR
        AC(I) = 0.
        BS(I) = 0.
        DO 4500 J = 1,NDPTS
          OMEGA = 2. * PI * (FLOAT(J-1) / FLOAT(NDPTS)) * (FREQEN(I)
     A                   * SDTIME * NDPTS)
          AC(I) = AC(I) + CSTR(J) * COS(OMEGA)
          BS(I) = BS(I) + CSTR(J) * SIN(OMEGA)
 4500   CONTINUE
        AC(I) = 2. * AC(I)
        BS(I) = 2. * BS(I)
      WRITE(6,9014) I, FREQEN(I), AC(I),BS(I)
 4000 CONTINUE
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C * RECONSTRUCT PERIODIC WAVEFORM FROM MAJOR
C       FREQUENCIES.
C       CALCULATE PER(I) FOR POINT I.
C
      DO 5000 J = 1,NDPTS
        PER(J) = 0.
        DO 7000 K = 1,NFR
          OMEGA = 2. * PI * (FLOAT(J-1) / FLOAT(NDPTS)) * (FREQEN(K)
     A                   * SDTIME * NDPTS)
          PER(J) = PER(J) + AC(K) * COS(OMEGA) + BS(K) * SIN(OMEGA)
 7000   CONTINUE
        PER(J) = PER(J) / FLOAT(NDPTS)
        PER(J) = PER(J) + A0 / (2. * FLOAT(NDPTS))
 5000 CONTINUE
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C * NOISE ARRAY IS LEVELED CURVE MINUS PERIODIC FIT TO IT.
C
      DO 6000 I = 1,NDPTS
        DSTR(I) = CSTR(I) - PER(I)
        SUBSTR(I) = DSTR(I)
      SUMER = SUMER + DSTR(I)
 6000 CONTINUE
      AVEER = SUMER / NDPTS
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C * PRINT ARRAYS OF ORIGINAL DATA, ORIGINAL ESTIMATE, DIFFERENCE,
C     POLYNOMIAL FIT, DIFFERENCE, PERIODIC FIT, AND DIFFERENCE (NOISE)
C
      WRITE(6,9016)
      WRITE(6,9003)
      DO 8000 I = 1,NDPTS
        WRITE(6,9004) I, SSTIME(I), BSTR(I), POLY(I),
     *                 CSTR(I), PER(I), DSTR(I)
 8000 CONTINUE
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
```

PROGRAM LISTING (Continued)

```
C * CALCULATE STANDARD DEVIATION OF NOISE
C
      SIGMA = 0.
      DO 8300 I = 1,NDPTS
        SIGMA = SIGMA + (DSTR(I) - AVEER) **2.
 8300 CONTINUE
      SIGMA = SQRT(SIGMA / FLOAT(NDPTS))
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C * NORMALIZE FIRST DERIVATIVE
C
      IIND = (SSTIME(1) - STIME(1)) / DTIME + 1
      IF (STIME(IIND) .NE. SSTIME(1)) CALL PERROR(64)
C
C * AVERAGE IN AVED,  STANDARD DEVIATION IN SIGD
C
      SUMD = 0.0
      DO 7400 I = 1,NSSPTS
        J = IIND + I - 1
        SUMD = STRDOT(J) + SUMD
 7400 CONTINUE
      AVED = SUMD / FLOAT(NSSPTS)
      SIGD = 0.
      DO 7600 I = 1,NSSPTS
        J = IIND + I - 1
        SIGD = SIGD + (STRDOT(J) - AVED) ** 2
 7600 CONTINUE
      SIGD = SQRT(SIGD / FLOAT(NSSPTS))
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C * NORMALIZE NOISE AND ANALYZE IT
C
      WRITE(6,9015)
      DO 8450 I = 1,NDPTS
        RNOISE(I) = (DSTR(I) - AVEER) / SIGMA
 8450 CONTINUE
      CALL GFIT(NORM,NBIN,RNOISE,NSSPTS,CELLS,COMP,CS,IDF,Q,IER)
      IF (IER .GT. 0) CALL PERROR(85)
      WRITE(6,9005) AVEER , SIGMA
      WRITE(6,9007) (CELLS(I),I = 1,NBIN)
      WRITE(6,9008) (COMP(I), I = 1,NBIN)
      WRITE(6,9006) IDF , CS , Q
      WRITE(6,9017)
      J = 0
      DO 8400 I = 1,NDPTS
        STAR = 1H
        IF (RNOISE(I) .GT. 3.0) STAR = 1H*
        SIGNDD = (STRDOT(I) - AVED) / SIGD
        IF (RNOISE(I) .LT. 2.5 .AND. SIGNDD .LT. 3.0) GO TO 8400
        WRITE(6,9018) I,SSTIME(I),DSTR(I),RNOISE(I),STAR,SIGNDD
 8400 CONTINUE
C
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C * CALCULATE INDEX IN STRING CORRESPONDING TO START OF ARRAY SUBSTR
C
```

PROGRAM LISTING (Continued)

```
      TIME = SSTIME(1)
      IF (SDTIME .NE. DTIME) CALL PERROR (63)
      IIND = (TIME - STIME(1)) / DTIME + 1
      IF (STIME(IIND) .NE. SSTIME(1)) CALL PERROR (64)
C
C * REPLACE SUBSTR BY ITS THEORETICAL APPROXIMATION OF A
C       POLYNOMIAL PLUS A PERIODIC.
C
      DO 5500 I = 1,NSSPTS
        IREP = IIND + I - 1
        REPSTR(IREP) = POLY(I) + PER(I)
 5500 CONTINUE
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C * SET IFREP TO INDICATE PROGRESS.
C
      IF (IFREP .EQ. 2) IFREP = 3
      IF (IFREP .EQ. 1) IFREP = 2
      IF (IFREP .EQ. 0) IFREP = 1
C
C***********************************************************************
C
      IF (PLOTOP(1) .EQ. 0) GO TO 8900
C
C                 PLOT  DATA SUBSET, POLYNOMIAL FIT
C
      DO 8500 I = 1,NDPTS
        YARR1(I) = BSTR(I)
        YARR2(I) = POLY(I)
        YARR3(I) = 0
        XARR(I) = SSTIME(I)
 8500 CONTINUE
      DELTAX = SDTIME
      NPOINT = NDPTS
      NARR = 2
      NPPI = 40
      CALL OVERLAY(1HJ,5,3)
C
C                 PLOT  DATA SUBSET - POLYNOMIAL FIT , PERIODIC FIT
C
      NARR = 2
      DO 8750 I = 1,NDPTS
        YARR1(I) = CSTR(I)
        YARR2(I) = PER(I)
        YARR3(I) = 0.
 8750 CONTINUE
      CALL OVERLAY(1HJ,5,3)
C
C                PLOT  NOISE = DATA SUBSET - POLYNOMIAL FIT -
C                                  PERIODIC FIT
C
      NARR = 1
      DO 8800 I = 1,NDPTS
        YARR1(I) = DSTR(I)
        YARR2(I) = 0.
        YARR3(I) = 0.
```

PROGRAM LISTING (Continued)

```
 8800 CONTINUE
      CALL OVERLAY(1HJ,5,3)
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
 8900 CONTINUE
      IF (IFREP .NE. NDREP) GO TO 9000
      IF (PLOTOP(2) .EQ. 0) GO TO 8990
C
C * PLOT DATA RECORD WITH NOISE REMOVED
C
      IF (IANOPT .EQ. 2) GO TO 8920
C
C         THEFFT CALLED FROM KITTY8
C         PLOT ONLY CURRENT DATA SET
C
        ISTART = IIND
        ISTOP  = IIND + NSSPTS - 1
        NPOINT = NSSPTS
        GO TO 8950
 8920 CONTINUE
C
C          THEFFT CALLED FROM COMPLT
C          PLOT ENTIRE DATA SET
C
        ISTART = 1
        ISTOP  = NSTPTS
        NPOINT = NSTPTS
 8950 CONTINUE
      NARR = 1
      DO 6500 I = ISTART, ISTOP
          XARR(I) = STIME(1)
          YARR1(I) = REPSTR(I)
          YARR2(I) = 0.
          YARR3(I) = 0.
 6500   CONTINUE
        CALL OVERLAY(1HJ,5,3)
 8990 CONTINUE
C
C*************************************************************************
C
 9000 RETURN
 9002 FORMAT(2I2)
 9003 FORMAT (1X,///,1X,5HINDEX,5X,4HTIME,1X,5X,6HSTRING,5X,
     *          4X,8HPOLY FIT,4X,2X,13HLEVELED CURVE,1X,
     *          4X,8HPERIODIC,4X,5X,5HNOISE,//)
 9004 FORMAT (1X,I5,F10.5, 7(2X,E14.7))
 9005 FURMAT(1X,///,10X,13HMEAN OF NOISE,17X,F10.3,/,
     A         10X,27HSTANDARD DEVIATION OF NOISE,3X,F10.3,/)
 9006 FORMAT (1H0,10X,18HDEGREES OF FREEDOM,31X,I10,/,
     *          15X,21HCHI-SQUARED STATISTIC,28X,F10.3,/,
     *          15X,21HCHI-SQUARED STATISTIC,/,
     *          15X,5X,39HEXCEEDING CS IF NULL HYPOTHESIS IS TRUE,5X,F10.3
     *          //)
 9007 FORMAT (10X,8HCELLS = ,10F10.3)
 9008 FORMAT (10X,8HCOMP =  ,10F10.3)
 9009 FORMAT(1X,////,10X,25HPLOT OPTION FOR THEFFT IS,5X,2I5)
 9011 FORMAT(1X,///,15X,54HCOEFFICIENTS OF FITTED POLYNOMIAL (C1, C2,...
     *, CN, CO),/,10X,8E15.7,//)
```

PROGRAM LISTING (Continued)

```
9012 FORMAT(1H1,///,20X,54(1H*),/,20X,1H*,52X,1H*,/,20X,1H*,13X,
     *        27HREGIME FITTING AND ANALYSIS,12X,1H*,/,20X,1H*,
     *        52X,1H*,/,20X,54(1H*))
9013 FORMAT(1X,/////,32X,9HFREQUENCY,16X,6HCOSINE,18X,4HSINE,
     *        /,35X,5H(KHZ),16X,11HCOEFFICIENT,14X,
     *        11HCOEFFICIENT,//)
9014 FORMAT(10X,I10, 3(10X,E14.7))
9015 FORMAT(1H1,///,20X,54(1H*),/,20X,1H*,52X,1H*,/,20X,
     *        1H*,18X,17HANALYSIS OF NOISE,17X,1H*,/,20X,
     *        1H*,52X,1H*,/,20X,54(1H*))
9016 FORMAT(1H1,///,20X,54(1H*),/,20X,1H*,52X,1H*,/,20X,1H*,
     *        20X,13HFITTED CURVES,19X,1H*,/,20X,1H*,52X,1H*,
     *        /,20X,54(1H*))
9017 FORMAT(1X,/////,31X,29HNOISE GREATER THAN 2.5 SIGMAS,/,
     *        6X,5HINDEX,8X,4HTIME,12X,5HNOISE,6X,10HNO. SIGMAS,
     *        5X,10HGT 3 SIGMA,5X,22HSIGMAS OF FIRST DERIV.,//)
9018 FORMAT(6X,I5,5X,F10.4,5X,F10.4,5X,F10.4,10X,A1,10X,F10.4)
9019 FORMAT(1H1,///,20X,54(1H*),/,20X,1H*,52X,1H*,/,20X,1H*,19X,
     *        14HPERIODIC CURVE,19X,1H*,/,20X,1H*,52X,
     *        1H*,/,20X,54(1H*))
     END
```

PROGRAM LISTING (Continued)

```
      SUBROUTINE FORIT (N, M, IER)
C-----------------------------------------------------------------------
C
C * SUBROUTINE FORIT DOES A FOURIER ANALYSIS OF A TABULATED
C     FUNCTION AND RETURNS THE FOURIER COEFFICIENTS.
C * IBM SUBROUTINE PACKAGE
C
C-----------------------------------------------------------------------
      COMMON / LEVELA / A(20), B(20), FNT(1500)
      COMMON /FORCOM/ AN,COEF,CONST,S1,C1,C,S,J,FNTZ,U1,U2,I,U0,Q
      LEVEL 2, N,M,IER,Q,U0,U1,U2,I,FNTZ,J,S,C,C1,S1,CONST,COEF,AN
C
      LEVEL 2, FNT, A, B
C
C * CHECK FOR PARAMETER ERRORS
C
      IER = 0
   20 IF (M) 30,40,40
   30 IER = 2
      RETURN
   40 IF (M-N) 60,60,50
   50 IER = 1
      RETURN
C
C * COMPUTE AND PRESET CONSTANTS
C
   60 AN = N
      COEF = 2.0 / (2.0 * AN + 1.0)
      CONST = 3.141593 * COEF
      S1 = SIN(CONST)
      C1 = COS(CONST)
      C = 1.0
      S = 0.0
      J = 1
      FNTZ = FNT(1)
   70 U2 = 0.0
      U1 = 0.0
      I = 2 * N + 1
C
C * FORM FOURIER COEFFICIENTS RECURSIVELY
C
   75 U0 = FNT(I) + 2.0 * C * U1 - U2
      U2 = U1
      U1 = U0
      I = I - 1
      IF (I-1) 80,80,75
   80 A(J) = COEF * (FNTZ + C*U1 - U2)
      B(J) = COEF * S * U1
      IF (J-(M+1)) 90,100,100
   90 Q = C1 * C - S1 * S
      S = C1 * S + S1 * C
      C = Q
      J = J + 1
      GO TO 70
  100 A(1) = A(1) * 0.5
      RETURN
      END
```

PROGRAM LISTING (Continued)

```
      SUBROUTINE NORM (X,P)
C-----------------------------------------------------------------------
C * NORM CALLS THE IMSL SUBROUTINE WHICH CALCULATES THE
C     GAUSSIAN DISTRIBUTION FOR A VALUE X
C
C-----------------------------------------------------------------------
      CALL MDNOR (X,P)
      RETURN
      END
```

PROGRAM LISTING (Continued)

```
      OVERLAY(H,5,1)
      PROGRAM POLYOR
C-----------------------------------------------------------------
C
C * REGRESSION ANALYSIS ON Y WITH RESPECT TO X,
C     POLYNOMIAL (DEGREE NCOEF-1), COEFFICIENTS IN
C     COEF, PREDICTION IN YPRED.
C
C-----------------------------------------------------------------
      COMMON / SSDATA / NSSPTS , SDTIME , SUBSTR(5000) , SSTIME(5000)
      COMMON / POLYCM / Y(1500) , MAXDEG , YPRED(1500) , COEF(10) ,NCOEF
      COMMON / CMPOLY / I,XYW(1500,7),MDP(3),NDMAX,NPT,RSQ,ALBP(2),
     *                  ANOVA(13),B(7,12),IB,PRED(1500,6),X(1500),IP,WK
      DOUBLE PRECISION WK(6000)
      LEVEL 2, Y, YPRED, MAXDEG, COEF, NCOEF, NSSPTS, SDTIME, SUBSTR,
     *         SSTIME , XYW, MDP, NDMAX, NPT, RSQ, ALBP, ANOVA, B, IP,
     *         PRED, X, I, WK
      LEVEL 2,IB
C
      DO 1000 I = 1,NSSPTS
        X(I) = SSTIME(I)
        XYW(I,1) = X(I)
        XYW(I,2) = Y(I)
        XYW(I,3) = 1.
 1000 CONTINUE
      NDMAX = 1500
      RSQ = 100.
      IB = 7
      IP = 1500
      MDP(1) = MAXDEG
      MDP(3) = 1
      ALBP(1) = 0.05
      ALBP(2) = 0.05
      NPT = NSSPTS
C
C * OBTAIN FITTED POLYNOMIAL USING ORTHOGONAL POLYNOMIALS
C
      CALL RLFOR (XYW,NDMAX,NPT,RSQ,MDP,ALBP,ANOVA,B,IB,PRED,
     *            IP,WK,IER)
      IF (IER .EQ. 33) CALL PERROR (87)
      IF (IER .GT. 0)  CALL PERROR (88)
      NCOEF = MDP(2) + 1
      DO 2000 I = 1,NCOEF
        COEF(I) = B(I,2)
 2000 CONTINUE
      DO 3000 I = 1,NSSPTS
        YPRED(I) = COEF(1)*X(I) + COEF(2)*X(I)**2 + COEF(3)*X(I)**3 +
     *             COEF(4)*X(I)**4 + COEF(5)
 3000 CONTINUE
      RETURN
      END
```

PROGRAM LISTING (Continued)

```
      OVERLAY(I,5,2)
      PROGRAM PAWS
C-----------------------------------------------------------------------
C
C * SUBROUTINES PAWS CALCULATES THE POWER SPECTRUM FOR ARRAY SUBSTR
C     USING SYSTEM SUBROUTINE FTFREQ
C     CALLED BY SUBROUTINE THEFFT
C     PAWS = POWER ANALYSIS OF WAVEFORM SPECTRA
C
C---------------------------------------------------FALCON R&D  NEA 6-5-7
      COMMON / CONTRL / LUIN , LUOUT , LUDATA , LURUN , LUREPT,
     *                  IFSVRP , ONLINE , IANOPT , OPTION , IFBACH,
     *                  USRNAM , PURPOS
      COMMON / SSDATA / NSSPTS , SDTIME , SUBSTR(5000) , SSTIME(5000)
      COMMON / PAWCOM / XYMV(6), ACV(1602), FREQ(801), PS(1602),
     *                  XCOV(1602), XSPECT(1602), AMPHAS(1602),
     *                  XFER(1602), COHER(801)
      COMMON /FREQPS/ A(810) , B(810) , R(1620) , NROW
      COMMON /FREQS/ FREQEN(20) , NFR
      COMMON / SCRIB2 / XARR(100) , YARR(100) , DX1 , NDPTS , NPPI
      COMMON / LAGCOM / M
      COMMON / PAWVAR / X(1500) , IND(6) , XIND(2) , PLOTOP,I,NFREQ
C
      INTEGER ONLINE , OPTION , USRNAM(2) , PURPOS(6) , PLOTOP
C
      LEVEL 2, NSSPTS, SDTIME, SUBSTR, SSTIME, A, B, R, FREQEN, NFR
     *       , LUIN,LUOUT,LUDATA,LURUN,LUREPT,IFSVRP,ONLINE,IANOPT
     *       , OPTION,USRNAM,PURPOS,M,NROW
     *       , IFBACH,XYMV,ACV,FREQ,PS,XCOV,XSPECT,AMPHAS,XFER,COHER
     *       , X,IND,XIND,PLOTOP,I,NFREQ
C
C * WRITE PAGE HEADING FOR PAWS
C
      WRITE(6,9005)
C
C
C * PLOTOP = PLOTTING OPTION FOR POWER SPECTRUM
C            = 0   OFF
C            = 1   ON
C
      READ(5,9003)  PLOTOP
      WRITE(6,9004) PLOTOP
C
C * PARAMETERS
C        M = NUMBER OF LAG TIMES
C
      M = 2. * (1. / SDTIME)
      IF (M .GT. NSSPTS) M = M / 2
      IF (M .GT. NSSPTS) M = M / 2
C
C * INITIALIZE FTFREQ ARRAYS
C
      DO 1000 I=1,NSSPTS
        X(I) = SUBSTR(I)
 1000 CONTINUE
      IND(1)  = 0
      IND(2)  = NSSPTS
```

PROGRAM LISTING (Continued)

```
      IND(3)   = 0
      IND(4)   = M
      IND(5)   = 0
      IND(6)   = 0
      XIND(1) = SUTIME
      XIND(2) = 0
C
C * CHECK INPUT TO SUBROUTINE FTFREQ
C
      IF (IND(2) .LE. 3) CALL PERROR(90)
      IF (IND(4) .LE. 2) CALL PERROR(91)
      IF (IND(4) .GE. IND(2)) CALL PERROR(92)
C
C * DO SPECTRUM ANALYSIS
C
      CALL FTFREQ (X, IND, XIND, XYMV, ACV, FREQ, PS,
     *             XCOV, XSPECT, AMPHAS, XFER, COHER, IER)
      IF (IER .NE. 0) WRITE (LUOUT, 9001) IER
C
C * PRINT OUTPUT FROM POWER SPECTRUM
C
      CALL PAWOUT
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C * PUT TOP 10 MAJOR FREQUENCIES (FREQUENCIES WITH LARGEST
C        POWER SPECTRUM) INTO FREQEN.
C
      NFR = 10
      NROW = M + 1
      DO 3000 I = 1,NROW
        A(I) = FREQ(I)
        B(I) = PS(I)
 3000 CONTINUE
C
C * SORT FREQUENCIES INTO R BY DECREASING POWER
C
      CALL ROWSRT(A,B,R,NROW)
C
C * PUT 10 LARGEST (FIRST 10) INTO FREQEN
C
      DO 4000 I = 1,NFR
        FREQEN(I) = R(I)
 4000 CONTINUE
C
C * WRITE 10 PREDOMINANT FREQUENCIES
C
      WRITE(6,9002) (FREQEN(I),I=1,NFR)
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      IF (PLOTOP .EQ. 0) GO TO 9000
C
C * PLOT LOG(POWER)VS FREQUENCY (FIRST 25)
C
      NFREQ = M + 1
      DO 5000 I = 1,NFREQ
        IF (PS(I) .LT. 1) PS(I) = 1.
        PS(I) = ALOG10(PS(I))
```

PROGRAM LISTING (Continued)

```
 5000 CONTINUE
      DX1 = FREQ(2) - FREQ(1)
      NDPTS = MIN0(NFREQ,IFIX(10. / DX1 + 1.))
      DO 6000 I = 1,NDPTS
        XARR(I) = FREQ(I)
        YARR(I) = PS(I)
 6000 CONTINUE
      NPPI = IFIX(1. / DX1)
      CALL SCRIB2
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
 9000 RETURN
 9001 FORMAT(1X,/,10X,46H**WARNING** ERROR TERM FROM SUBROUTINE FTFREQ
    A        26H(DEFINED BY SUBROUTINE) IS, I5, //)
 9002 FORMAT (1X,///,20X,27HTEN PREDOMINANT FREQUENCIES,
    A        //,5X,10F10.5,/)
 9003 FORMAT(I2)
 9004 FORMAT(1X,////,10X,,23HPLOT OPTION FOR PAWS IS,5X,I5)
 9005 FORMAT(1H1,///,20X,54(1H*),/,20X,1H*,52X,1H*,/,20X,1H*,15X,
    *        23HPOWER SPECTRUM ANALYSIS,14X,1H*,/,20X,1H*,52X,1H*,/,
    *        20X,1H*,52X,1H*,/,20X,54(1H*))
      END
```

PROGRAM LISTING (Continued)

```
      SUBROUTINE PAWOUT
C---------------------------------------------------------------------
C
C * SUBROUTINE PAWOUT PRINTS THE OUTPUT FROM SUBROUTINE PAWS
C
C-------------------------------------------FALCON R&D  NEA 6-5-79
      COMMON / CONTRL / LUIN , LUOUT , LUDATA , LURUN , LUREPT
      COMMON / CONTRL / IFSVRP , ONLINE , IANOPT , OPTION , IFBACH
      COMMON / CONTRL / USRNAM , PURPOS
      INTEGER ONLINE , OPTION , USRNAM(2) , PURPOS(6)
      COMMON / PAWCOM / XYMV(6), ACV(1602), FREQ(801), PS(1602),
     *                  XCOV(1602), XSPECT(1602), AMPHAS(1602),
     *                  XFER(1602), COHER(801)
      LEVEL 2, LUIN,LUOUT,LUDATA,LURUN,LUREPT,IFSVRP,ONLINE,IANOPT
     *       , OPTION,USRNAM,PURPOS,M,LAG,FIND
     *       , XYMV,ACV,FREQ,PS,XCOV,XSPECT,AMPHAS,XFER,COHER
      LEVEL 2,IFBACH
      COMMON /PCOMM/ LAG , FIND
      COMMON / LAGCOM / M
C
C * PRINT HEADING, MEAN, VARIANCE
C
      WRITE (LUOUT, 9001)
      WRITE (LUOUT, 9002) XYMV(1), XYMV(2)
C
C * PRINT LAG AND AUTOCOVARIANCE (LAG) TABLE
C         FREQUENCY AND POWER SPECTRUM TABLE
C
      WRITE (LUOUT, 9003)
      LAG = 0
      WRITE (LUOUT, 9004) LAG, XYMV(2), FREQ(1), PS(1)
      DO 1000 LAG=1,M
        FIND = LAG + 1
        WRITE (LUOUT, 9004) LAG, ACV(LAG), FREQ(FIND), PS(FIND)
 1000 CONTINUE
      WRITE (LUOUT, 9005)
 9000 RETURN
 9001 FORMAT (1H1, //, 20X, 34HRESULTS OF POWER SPECTRUM ANALYSIS, // )
 9002 FORMAT (1X, 10X, 13HWAVEFORM MEAN, F15.4, /, 11X,
     *        17HWAVEFORM VARIANCE, F15.4, /)
 9003 FORMAT (1X, //, 17X, 20HAUTOCOVARIANCE TABLE, 10X,
     *        14HPOWER SPECTRUM, //, 17X, 3HLAG, 3X, 14HAUTOCOVARIANCE,
     *        5X, 9HFREQUENCY, 3X, 14HPOWER SPECTRUM, /)
 9004 FORMAT (10X, I10, F15.5, 5X, F10.5, F15.5)
 9005 FORMAT (1X, ///, 25X, 30HEND OF POWER SPECTRUM ANALYSIS, ////)
      END
```

PROGRAM LISTING (Continued)

```
      SUBROUTINE ROWSRT(A,B,R,N)
C-----------------------------------------------------------------------
C
C * SUBROUTINE ROWSRT SORTS ROWS OF A MATRIX A ACCORDING TO DESCENDING
C      ORDER OF MATRIX B.  STORE SORTED A INTO R.
C
C-----------------------------------------------------------------------
      DIMENSION A(810), B(810), R(1620)
      COMMON /SRTCOM/ I,I2,ISORT,RSAVE,SAVER,IR,IA
      LEVEL 2, A,B,R,I,N,I2,ISORT,RSAVE,SAVER,IR,IA
C
C * MOVE SORTING KEY VECTOR TO FIRST COLUMN OF OUTPUT MATRIX AND
C      BUILD ORIGINAL SEQUENCE LIST IN SECOND COLUMN
C
      DO 1000 I = 1,N
        R(I) = B(I)
        I2 = I + N
        R(I2) = I
 1000 CONTINUE
C
C * SORT ELEMENTS IN SORTING KEY VECTOR (ORIGINAL SEQUENCE LIST IS
C      RESEQUENCED ACCORDINGLY)
C
 2000 CONTINUE
      ISORT = 0
      DO 4000 I = 2,N
        IF (R(I) - R(I-1)) 4000, 4000, 3000
 3000   CONTINUE
        ISORT = ISORT + 1
        RSAVE = R(I)
        R(I) = R(I-1)
        R(I-1) = RSAVE
        I2 = I + N
        SAVER = R(I2)
        R(I2) = R(I2-1)
        R(I2-1) = SAVER
 4000 CONTINUE
      IF (ISORT) 2000, 5000, 2000
C
C * MOVE ROWS FROM MATRIX A TO MATRIX R (NUMBER IN SECOND COLUMN OF R
C      REPRESENTS ROW NUMBER OF MATRIX A TO BE MOVED)
C
 5000 CONTINUE
      DO 8000 I = 1,N
C
C * MOVE ELEMENT TO OUTPUT MATRIX
C
        I2 = I + N
        R(I) = A( R(I2) )
 8000 CONTINUE
      RETURN
      END
```

PROGRAM LISTING (Continued)

```
      SUBROUTINE SCRIB2
C-----------------------------------------------------------------
C
C * SCRIB2 PLOTS GRAPHS FOR MUZZLEDEVICE DATA IN PAWS
C       USING CALCOMP PLOTTING PACKAGE
C
C-----------------------------------------------------------------
      COMMON / SCRIB2 / XARRAY(100) , YARRAY(100) , DX1 , NDPTS , NPPI
      REAL       DX , XSCALE , YSCALE ,XLNGTH
      REAL       YLNGTH , XOR , YOR , XBASE , YBASE , FACTOR
      INTEGER     ISENT(10) , LABEL(4) , IUNIT
      COMMON / LABEL / KTB(24) , TSTART , I
      LEVEL 2, KTB , TSTART , I
C
      DATA LABEL/'BOOTS','B390','X3983','M392 DATA'/
C
C                TEST ARGUMENTS FOR ERRORS
C
      IF (NDPTS .GT. 100) CALL PERROR (93)
C
C                DEFINE VARIABLES USED BY PLOTTING ROUTINES
C
      XLNGTH=11.
      YLNGTH=8.5
      XSIZE=8.0
      YSIZE=6.0
      XBASE=2.0
      YBASE=1.0
      CALL FIXSCA(XARRAY(1),NDPTS,XSIZE,XSCALE,XOR,XMAX,DX)
      CALL FIXSCA(YARRAY(1),NDPTS,YSIZE,YSCALE,YOR,YMAX,DY)
      XLNGTH=11.
      YLNGTH=8.5
      XSIZE=8.0
      YSIZE=6.0
      XBASE=2.0
      YBASE=1.0
      CALL FIXSCA(XARRAY(1),NDPTS,XSIZE,XSCALE,XOR,XMAX,DX)
      CALL FIXSCA(YARRAY(1),NDPTS,YSIZE,YSCALE,YOR,YMAX,DY)
      FACTOR = 1.
      IUNIT  = 10
C
C                BRL PLOTTING ROUTINES (INITIAL)
C
      CALL PLTBEG(XLNGTH,YLNGTH,FACTOR,IUNIT,LABEL)
      CALL PLTSCA(XBASE,YBASE,XOR,YOR,XSCALE,YSCALE)
C
C                PLOT THE CURVES
C
        CALL PLTDTS(1,0,XARRAY,YARRAY,NDPTS,0)
C
C                PLOT AXES AND LABELS
C
      MODE = 4
      ONE = 1.
      CALL PLTAXS(DX,DY,XOR,XMAX,YOR,YMAX,MODE)
      CALL LABELA(DX,DY,XOR,XMAX,YOR,YMAX,ONE,ONE)
C
```

PROGRAM LISTING (Continued)

```
      YMAX1=YMAX+1.0*YSCALE
      YMAX1=YMAX+1.0*YSCALE
      ENCODE (100, 130, ISENT)   KTB
  130 FORMAT (1X, 24I2, '>')
      CALL PLTSYM(0.15,ISENT,0.,XOR,YMAX1)
      CALL PLTSYM(0.15,ISENT,0.,XOR,YMAX1)
C
      CALL PLTPGE
      RETURN
      END
```

PROGRAM LISTING (Continued)

```
      OVERLAY(J,5,3)
      PROGRAM SCRIBL
C------------------------------------------------------------------
C
C * SCRIBL PLOTS UP TO THREE OVERLAPPING GRAPHS FOR
C       MUZZLEDEVICE DATA USING CALCOMP PLOTTING PACKAGE
C
C------------------------------------------------------------------
      COMMON / SCRIB  / XARRAY(5000) , YARRY1(5000) , YARRY2(5000)
     A               YARRY3(5000) , NARRAY , DX1 , NDPTS , NPPI
      LOGICAL TWICE
      REAL       DX , XSCALE , YSCALE ,XLNGTH
      REAL       YLNGTH , XOR , YOR , XBASE , YBASE , FACTOR
      INTEGER     ISENT(10) , LABEL(4) , IUNIT
      COMMON / LABEL / KTB(24) , TSTART , I
      LEVEL 2, KTB, TSTART, I
C
      DATA LABEL/'BOOTS','B390','X3983','M392 DATA'/
C
C                TEST ARGUMENTS FOR ERRORS
C
      IF (NDPTS .GT. 5000) CALL PERROR (84)
      IF ( (NARRAY .LT. 1) .OR. (NARRAY .GT. 3) ) CALL PERROR (83)
C
C                DEFINE VARIABLES USED BY PLOTTING ROUTINES
C
      XLNGTH=11.
      YLNGTH=8.5
      XSIZE=8.0
      YSIZE=6.0
      XBASE=2.0
      YBASE=1.0
      CALL FIXSCA(XARRAY(1),NDPTS,XSIZE,XSCALE,XOR,XMAX,DX)
      CALL FIXSCA(YARRY1(1),NDPTS,YSIZE,YSCALE,YOR,YMAX,DY)
      XLNGTH=11.
      YLNGTH=8.5
      XSIZE=8.0
      YSIZE=6.0
      XBASE=2.0
      YBASE=1.0
      CALL FIXSCA(XARRAY(1),NDPTS,XSIZE,XSCALE,XOR,XMAX,DX)
      CALL FIXSCA(YARRY1(1),NDPTS,YSIZE,YSCALE,YOR,YMAX,DY)
      FACTOR = 1.
      IUNIT  = 10
      TWICE = .FALSE.
C
C                BRL PLOTTING ROUTINES (INITIAL)
C
      CALL PLTBEG(XLNGTH,YLNGTH,FACTOR,IUNIT,LABEL)
      CALL PLTSCA(XBASE,YBASE,XOR,YOR,XSCALE,YSCALE)
C
C                PLOT THE CURVES
C
      GO TO (3000,2000,1000) , NARRAY
 1000 CONTINUE
        CALL PLTDTS(1,0,XARRAY,YARRY3,NDPTS,0)
 2000 CONTINUE
```

PROGRAM LISTING (Continued)

```
          CALL PLTDTS(4,0,XARRAY,YARRY2,NDPTS,0)
 3000 CONTINUE
          CALL PLTDTS(1,0,XARRAY,YARRY1,NDPTS,0)
C
C                        PLOT AXES AND LABELS
C
      MODE = 4
      ONE = 1.
      CALL PLTAXS(DX,DY,XOR,XMAX,YOR,YMAX,MODE)
      CALL LABELA(DX,DY,XOR,XMAX,YOR,YMAX,ONE,ONE)
C
      YMAX1=YMAX+1.0*YSCALE
      YMAX1=YMAX+1.0*YSCALE
      ENCODE (100, 130, ISENT)  KTB
  130 FORMAT (1X, 24I2, '>')
      CALL PLTSYM(0.15,ISENT,0.,XOR,YMAX1)
      CALL PLTSYM(0.15,ISENT,0.,XOR,YMAX1)
C
C     IF (TWICE) GO TO 500
      CALL PLTPGE
C 500 CONTINUE
      TWICE = .TRUE.
      RETURN
      END
```

PROGRAM LISTING (Continued)

```
      OVERLAY(G,6,0)
      PROGRAM SMOOTH
C-------------------------------------------------------------------
C
C * SUBROUTINE SMOOTH ALLOWS THE USER TO SELECT A
C     SMOOTHING OPTION.
C
C------------------------------------FALCON R&D  MEK 7- 9-79
      COMMON / CONTRL / LUIN , LUOUT , LUDATA , LURUN , LUREPT
      COMMON/CONTRL/IFSVRP,ONLINE,IANOPT,OPTION,IFBACH
      COMMON / CONTRL / USRNAM , PURPOS
      COMMON / SSDATA / NSSPTS, SDTIME , SUBSTR(5000) , SSTIME(5000)
      INTEGER ONLINE , OPTION , USRNAM(2) , PURPOS(6)
      INTEGER SMOPT
      LEVEL 2, NSSPTS, SDTIME, SUBSTR, SSTIME
      LEVEL 2, LUIN,LUOUT,LUDATA,LURUN,LUREPT,IFSVRP,ONLINE,IANOPT
      LEVEL 2, OPTION,USRNAM,PURPOS,SMOPT,NSMO
      LEVEL 2,IFBACH
      COMMON / COMSMO / SMOPT , NSMO
C
C * USER SELECTS A SMOOTHING OPTION (BATCH)
C
      READ (LUIN, 9002) SMOPT
C
C * SELECT NUMBER OF POINTS FOR SMOOTHING
C
C
 2000 CONTINUE
      WRITE (6, 9003) SMOPT
      IF (SMOPT .EQ. 0) GO TO 9000
      IF (SMOPT .LT. 1 .OR. SMOPT .GT. 2) CALL PERROR(94)
      GO TO (3000, 4000), SMOPT
 3000 CONTINUE
C
C * 11-POINT MEDIAN - SMOOTHING OF SUBSTR
C
      CALL MED11
      GO TO 8000
 4000 CONTINUE
C
C * NSMO-POINT AVERAGE - SMOOTHING OF SUBSTR
C
      NSMO = 11
      CALL MOVAVE (SUBSTR, NSSPTS, NSMO)
      GO TO 8000
 8000 CONTINUE
 9000 RETURN
 9002 FORMAT (I2)
 9003 FORMAT (1X, 20HSMOOTHING OPTION IS , I2)
      END
```

PROGRAM LISTING (Continued)

```
      SUBROUTINE MED11
C---------------------------------------------------------------------
C
C * SUBROUTINE MED11
C     11-POINT MEDIAN - SMOOTHING OF SUBSTR
C
C-------------------------------------FALCON R&D  NEA 7- 9-79
      COMMON / SSDATA / NSSPTS , SDTIME , SUBSTR(5000) , SSTIME(5000)
      COMMON / COMM11 / L, Z, ZM, IU
      DIMENSION Z(11)
      COMMON / GENSTR / X(5000), LL, I, LS, LN, K, KS
      LEVEL 2, NSSPTS, SDTIME, SUBSTR, SSTIME, X
      LEVEL 2, LL, L, I, Z, ZM, IU, LS, LN, K, KS
      L = 11
      LL = .5 * L
      DO 1000 I = 1,L
        Z(I) = SUBSTR(I)
 1000 CONTINUE
      CALL MEDIAN(Z,L,ZM)
      DO 1500 I = 1,LL
        X(I) = ZM
 1500 CONTINUE
      DO 2000 I = 1,L
        IU = NSSPTS - I + 1
        Z(I) = SUBSTR(I)
 2000 CONTINUE
      CALL MEDIAN(Z,L,ZM)
      DO 2500 I = 1,LL
        IU = NSSPTS - I + 1
        X(IU) = ZM
 2500 CONTINUE
      LS = LL + 1
      LN = NSSPTS - LL
      DO 3000 K = LS,LN
        KS = K - LL - 1
        DO 4000 I = 1,L
          Z(I) = SUBSTR(KS + I)
 4000   CONTINUE
        CALL MEDIAN(Z,L,ZM)
        X(K) = ZM
 3000 CONTINUE
      DO 5000 I = 1,NSSPTS
        SUBSTR(I) = X(I)
 5000 CONTINUE
      RETURN
      END
```

PROGRAM LISTING (Continued)

```
      SUBROUTINE MEDIAN(Z,N,ZM)
C-----------------------------------------------------------------------
C
C * SUBROUTINE MEDIAN RETURNS THE MEDIAN ZM OF N NUMBERS
C      IN ARRAY Z
C
C-----------------------------------FALCON R&D  NEA 7- 9-79
      DIMENSION Z(N)
      COMMON / COMMED / L, J, AMIN, IMIN, I
      LEVEL 2, Z, N, ZM, L, J, AMIN,IMIN,I
      L = N / 2 + 1
      DO 1000 J = 1,L
        AMIN = 10000000.
        DO 2000 I = 1,N
          IF (Z(I) .GE. AMIN) GO TO 2000
            IMIN = I
            AMIN = Z(I)
 2000   CONTINUE
        IF (J .EQ. L) GO TO 3000
        Z(IMIN) = 1000000.
 1000 CONTINUE
 3000 CONTINUE
      ZM = Z(IMIN)
      RETURN
      END
```

PROGRAM LISTING (Continued)

```
      SUBROUTINE MOVAVE (S, N, L)
C-----------------------------------------------------------------
C
C * SUBROUTINE MOVAVE
C     L-POINT AVERAGE - SMOOTHING OF ARRAY S WITH N ELEMENTS
C
C-----------------------------------FALCON R&D  NEA 7- 9-79
      DIMENSION S(N)
      COMMON / COMMOV / SS, SL, IL, SA, SC
      COMMON / GENSTR / X(5000), LL, I, LS, LN, K, KS
      LEVEL 2, S, N, L, X
      LEVEL 2, LL,SS,SL,IL,I,SA,SC,LS,LN,K,KS
      LL= .5 * L
      SS = 0.
      SL = 0.
      DO 1000 I = 1,LL
        IL = N - I + 1
        SS = SS + S(I)
        SL = SL + S(IL)
 1000 CONTINUE
      SA = SS / FLOAT(LL)
      SC = SL / FLOAT(LL)
      DO 2000 I = 1,LL
        IL = N - I + 1
        X(I) = SA
        X(IL) = SC
 2000 CONTINUE
      LS = LL + 1
      LN = N - LL
      DO 4000 K = LS,LN
        X(K) = 0.
        KS = K - LL - 1
        DO 3000 I = 1,L
          X(K) = X(K) + S(KS + I)
 3000   CONTINUE
        X(K) = X(K) / FLOAT(L)
 4000 CONTINUE
      DO 5000 K = 1,N
        S(K) = X(K)
 5000 CONTINUE
      RETURN
      END
```

PROGRAM LISTING (Continued)

```
      OVERLAY(L,7,0)
      PROGRAM DERIV
C-----------------------------------------------------------------
C
C * SUBROUTINE DERIV CALCULATES THE FIRST AND SECOND
C      DERIVATIVES OF ENTIRE DATA STRING.  PLOTS DATA
C      STRING AND DERIVATIVES.
C
C-----------------------------------------------------------------
      COMMON / SDATA  / NSTPTS , DTIME , STRING(5000) , STIME(5000)
      COMMON / DERIVS / STRDOT(5000) , STRDD(5000) , PLOTOP
      COMMON / SCRIB4 / XARRAY(5000) , YARRY1(5000) , YARRY2(5000) ,
     *                  YARRY3(5000) , DX1 , NDPTS , NPPI
      COMMON / LABEL / KTB(24) , TSTART , I
      COMMON / CLASCM / ICLASS , NBEGBL
      COMMON / COMDER / NSTM2 , CD , DT , CDD
C
      LEVEL 2, NSTPTS, DTIME, STRING, STIME, STRDOT, STRDD, KTB,TSTART
     *          ,I,PLOTOP,ICLASS,NBEGBL,NSTM2,CD,DT,CDD
      INTEGER PLOTOP
C
C * WRITE PAGE HEADING FOR DERIVATIVES
C
      WRITE(6,9004)
C
C * PLOTOP = PLOT OPTION FOR DERIVATIVES
C          = 1  ON
C          = 0  OFF
C
      READ(5,9002)  PLOTOP
      WRITE(6,9003) PLOTOP
      DO 1000 I = 1,NSTPTS
        STRDOT(I) = 0.
$       STRDD(I) = 0.
 1000 CONTINUE
      DT = DTIME * .001
      CD = 1. / (12. * DT)
      CDD = CD / DT
      NSTM2 = NSTPTS - 2
C
C * ICLASS INDICATES SATURATION LEVEL:
C      ICLASS = -1, HEAVY SATURATION, EPOCH 1 DETECTION
C                   AND REGIME 1 ANALYSIS ONLY.
C      ICLASS =  0, NO SATURATION, FULL ANALYSIS.
C      ICLASS =  1, LIMITED SATURATION (CLIPPING),
C                   FULL ANALYSIS.
C
      IF (ICLASS .EQ. -1) NSTM2 = NBEGBL - 2
      DO 2000 I = 3,NSTM2
        STRDOT(I) = CD * (8. * (STRING(I+1) - STRING(I-1))
     *                   - STRING(I+2) + STRING(I-2))
$       STRDD(I) = CDD * (16. * (STRING(I+1) + STRING(I-1))
$    *                   - 30. * STRING(I) - STRING(I+2) - STRING(I-2))
 2000 CONTINUE
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
```

PROGRAM LISTING (Continued)

```
C * WRITE OUT FIRST AND SECOND DERIVATIVES
C
      WRITE(6,9005)
$     WRITE(6,9006)
$     DO 2500 I = 1,NSTPTS
$       WRITE(6,9001) I,STIME(I),STRING(I),STRDOT(I),STRDD(I)
$2500 CONTINUE
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      IF (PLOTOP .EQ. 0) GO TO 4000
C
      DO 3000 I = 1,NSTPTS
        XARRAY(I) = STIME(I)
        YARRY1(I) = STRING(I)
        YARRY2(I) = STRDOT(I)
$       YARRY3(I) = STRDD(I)
 3000 CONTINUE
      DX1 = DTIME
      NDPTS = NSTPTS
      NPPI = 80
      CALL OVERLAY(1HM,7,1)
 4000 CONTINUE
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
 9001 FORMAT(19X,I4,2X,F9.5,2X,F9.5,2X,E13.7,2X,E13.7)
 9002 FORMAT(I2)
 9003 FORMAT(1X,///,10X,24HPLOT OPTION FOR DERIV IS,5X,I2)
 9004 FORMAT(1H1,///,20X,54(1H*),/,20X,1H*,52X,1H*,/,20X,1H*,15X,
     *       22HDERIVATIVE CALCULATION,15X,1H*,/,20X,1H*,52X,
     *       1H*,/,20X,54(1H*))
 9005 FORMAT(1X,/,10X,29HFIRST AND SECOND DERIVATIVES ,
     *       31HARE CALCULATED BUT NOT PRINTED.)
 9006 FORMAT(1X,6(/),18X,5HPOINT,4X,4HTIME,7X,4HDATA,8X,
     *       5HFIRST,9X,6HSECOND,//)
      RETURN
      END
```

PROGRAM LISTING (Continued)

```
      OVERLAY(M,7,1)
      PROGRAM SCRIB4
C-----------------------------------------------------------------------
C
C * SCRIB4 PLOTS THREE GRAPHS OF THE ORIGINAL DATA STRING AND ITS
C      FIRST AND SECOND DERIVATIVES USING CALCOMP PLOTTING PACKAGE
C
C-----------------------------------------------------------------------
      COMMON / SCRIB4 / XARRAY(5000) , YARRY1(5000) , YARRY2(5000) ,
     A             YARRY3(5000) , DX1 , NDPTS , NPPI
      REAL DX1 , XSCALE , YSCALE , XLNGTH
      REAL YLNGTH , XOR , YOR , XBASE , YBASE , FACTOR
      INTEGER ISENT(10) , LABEL(4) , IUNIT
      COMMON / LABEL / KTB(24) , TSTART , I
      LEVEL 2, KTB , TSTART , I
C
      DATA LABEL/'BOOTS','B390','X3983','M392 DATA'/
C
C
C
      IF (NDPTS .GT. 5000) CALL PERROR(95)
C
C               DEFINE VARIABLES USED TO PLOT YARRY1
C
      MODE = 0
      ONE = 1.
C
      XLNGTH=11.
      YLNGTH=8.5
      XSIZE=8.0
      YSIZE1=4.
      YSIZE2=1.5
      XBASE=2.
      YBASE=3.0
      CALL FIXSCA(XARRAY(1),NDPTS,XSIZE,XSCALE,XOR,XMAX,DX)
      CALL FIXSCA(YARRY1(1),NDPTS,YSIZE1,YSCALE,YOR,YMAX,DY)
      XLNGTH=11.
      YLNGTH=8.5
      XSIZE=8.0
      YSIZE1=4.
      YSIZE2=1.5
      XBASE=2.
      YBASE=3.0
      CALL FIXSCA(XARRAY(1),NDPTS,XSIZE,XSCALE,XOR,XMAX,DX)
      CALL FIXSCA(YARRY1(1),NDPTS,YSIZE1,YSCALE,YOR,YMAX,DY)
      FACTOR = 1.
      IUNIT = 10
C
C             PLOT DATA STRING
C
      CALL PLTBEG(XLNGTH,YLNGTH,FACTOR,IUNIT,LABEL)
      CALL PLTSCA(XBASE,YBASE,XOR,YOR,XSCALE,YSCALE)
      CALL PLTDTS(1,0,XARRAY,YARRY1,NDPTS,0)
      CALL PLTAXS(DX,DY,XOR,XMAX,YOR,YMAX,MODE)
      CALL LABELA(0.,DY,XOR,XMAX,YOR,YMAX,ONE,ONE)
      CALL LABELA(0.,DY,XOR,XMAX,YOR,YMAX,ONE,ONE)
C
```

PROGRAM LISTING (Continued)

```
C               DEFINE VARIABLES USED TO PLOT YARRY2
C
      CALL FIXSCA(YARRY2(1),NDPTS,YSIZE2,YSCALE,YOR,YMAX,DY)
      YBASE=1.
      CALL FIXSCA(YARRY2(1),NDPTS,YSIZE2,YSCALE,YOR,YMAX,DY)
      YBASE=1.
C
C               PLOT FIRST DERIVATIVE
C
      CALL PLTSCA(XBASE,YBASE,XOR,YOR,XSCALE,YSCALE)
      CALL PLTDTS(1,0,XARRAY,YARRY2,NDPTS,0)
      CALL PLTAXS(DX,DY,XOR,XMAX,YOR,YMAX,MODE)
      CALL PLTAXS(DX,DY,XOR,XMAX,YOR,YMAX,MODE)
      CALL LABELA(DX,DY,XOR,XMAX,YOR,YMAX,ONE,ONE)
C
C               DEFINE VARIABLES USED TO PLOT YARRY3
C
$     CALL FIXSCA(YARRY3(1),NDPTS,6.,YS3,SMINDD,SMAXDD,YELD3)
$     YSCALE = YS3
$     DY = YELD3
$     YOR = SMINDD
$     YMAX = SMAXDD
$     YBASE = 22.
$     XMAX = XOR
C
C               PLOT SECOND DERIVATIVE
C
$     CALL PLTSCA(XBASE,YBASE,XOR,YOR,XSCALE,YSCALE)
$     CALL PLTDTS(1,0,XARRAY,YARRY3,NDPTS,0)
$     CALL PLTAXS(0.,DY,XOR,XMAX,YOR,YMAX,MODE)
$     CALL LABELA(DX,DY,XOR,XMAX,YOR,YMAX,ONE,ONE)
C
      YMIN1=YOR+(YSCALE*6.3)
      YMIN1=YOR+(YSCALE*6.3)
      ENCODE(100, 130, ISENT) KTB
  130 FORMAT(1X,24I2,'>')
      CALL PLTSYM(0.15,ISENT,0.,XOR,YMIN1)
      CALL PLTSYM(0.15,ISENT,0.,XOR,YMIN1)
      CALL PLTPGE
      RETURN
      END
```

PROGRAM LISTING (Continued)

```
      OVERLAY(N,10,0)
      PROGRAM DETECT
C-----------------------------------------------------------------------
C
C * SUBROUTINE DETECT TRIES TO LOCATE THE FIRST EPOCH
C       AND THE SECOND EPOCH.
C
C-----------------------------------------------------------------------
      COMMON / SDATA  / NSTPTS , DTIME , STRING(5000) , STIME(5000)
      COMMON / DERIVS / STRDOT(5000) , STRDD(5000) , PLOTOP
      COMMON / COMDET / A(810),B(810),R(1620),NROW,I,NMAXDV,STRMAX,IP
     *                , N(200), STR(200), STRMIN , BACK , CURR
      COMMON / DTECT  / EPOCH1 , EPOCH2
      COMMON / CLASCM / ICLASS , NBEGBL
      INTEGER    EPOCH1, EPOCH2 , PLOTOP
      LEVEL 2, A, B, R, NROW, NSTPTS, DTIME, STRING, STIME, STRDOT,
     *          STRDD, PLOTOP, I, NMAXDV, STRMAX, IP,
     *          N,STR,STRMIN,EPOCH1,EPOCH2,ICLASS,NBEGBL
      LEVEL 2,BACK,CURR
C
C * WRITE PAGE HEADING FOR DETECT
C
      WRITE(6,9003)
C
C * FIND THE MAXIMUM OF STRDOT
C
      STRMAX = -99999.
      DO 2000 I = 1,NSTPTS
        IF (STRDOT(I) - STRMAX .LE. 0) GO TO 1000
          STRMAX = STRDOT(I)
          NMAXDV = I
 1000   CONTINUE
 2000 CONTINUE
C***********************************************************************
C
C * SORT STRING
C
      NROW = 200
      DO 3000 I = 1,NROW
        A(I) = FLOAT( NMAXDV - (NROW - I))
        B(I) = STRING(NMAXDV - (NROW - I))
 3000 CONTINUE
      CALL ROWSRT(A,B,R,NROW)
C
C * FIND THE EPOCH
C
C * IN AN INTERVAL OF 200 POINTS PRECEDING MAX DERIV,
C      WORK BACKWARDS TO FIND LAST POINT GREATER THAN
C      POINTS PRECEDING IT.
C
      DO 4000 I = 1,NROW
        IP = NMAXDV - I + 1
        IF (ABS(R(I) - FLOAT(IP)) .GT. 4.) GO TO 5000
 4000 CONTINUE
 5000 CONTINUE
      EPOCH1 = IP + 1
C
```

PROGRAM LISTING (Continued)

```
C * WORK BACKWARDS TO FIND LAST MONOTONICALLY
C     DECREASING POINT.  THAT IS EPOCH1.
C
      CURR = EPOCH1
 5400 CONTINUE
      BACK = CURR - 1
      IF (BACK .LE. 1.) CALL PERROR(65)
      IF (STRING(BACK) .GE. STRING(CURR)) GO TO 5800
        CURR = BACK
        GO TO 5400
 5800 CONTINUE
C
      EPOCH1 = CURR
C**********************************************************************
C
C * ICLASS INDICATES SATURATION LEVEL:
C       ICLASS = -1, HEAVY SATURATION, EPOCH 1 DETECTION
C                    AND REGIME 1 ANALYSIS ONLY.
C       ICLASS =  0, NO SATURATION, FULL ANALYSIS.
C       ICLASS =  1, LIMITED SATURATION (CLIPPING),
C                    FULL ANALYSIS.
C
      IF (ICLASS .EQ. -1) GO TO 9000
C**********************************************************************
C
C * IDENTIFY FIRST 200 VALUES AFTER NMAXDV OF STRING
C
      NROW = 200
      DO 6000 I = 1,NROW
        N(I) = NMAXDV + (I - 1)
        STR(I) = STRING(N(I))
 6000 CONTINUE
C
C * FIND FIRST MINIMUM OF STR(I)
C
 6500 CONTINUE
      STRMIN = 9999999.
      IP = 0
      DO 8000 I = 1,NROW
        IF (STR(I) .GE. STRMIN) GO TO 7000
          STRMIN = STR(I)
          IP = N(I)
 7000   CONTINUE
 8000 CONTINUE
      EPOCH2 = IP
C
C * CHECK IF 3 PRECEDING POINTS TO EPOCH2 ARE
C     MONOTONICALLY DECREASING.
C
      CURR = EPOCH2
      DO 8300 I = 1,3
        BACK = CURR - 1
        IF (STRING(BACK) .LE. STRING(CURR)) GO TO 8500
        CURR = BACK
 8300 CONTINUE
C
C * EPOCH2 IS VALID
```

PROGRAM LISTING (Continued)

```
C
      GO TO 8700
 8500 CONTINUE
C
C * EPOCH2 IS INVALID, REPEAT PROCESS TO THAT POINT
C
      NROW = CURR - NMAXDV
      GO TO 6500
 8700 CONTINUE
C***********************************************************************
      WRITE(6,9001) EPOCH1,STIME(EPOCH1),STRING(EPOCH1)
      WRITE(6,9002) EPOCH2,STIME(EPOCH2),STRING(EPOCH2)
C***********************************************************************
C
C *    FIND LOCAL EXTREMA IN DERIVATIVE
C
      NST=NMAXDV-40
      IF(NST.LT.2)NST=2
      NSP=NST+120
      CR=STRMAX*.08
      EXTR=STRDOT(NST)
      WRITE(6,9006)
      DO 8900 I=NST,NSP
      A1=STRDOT(I-1)
      A2=STRDOT(I)
      A3=STRDOT(I+1)
      IF(A1.LE.A2.AND.A2.GE.A3.AND.ABS(A2-EXTR).GE.CR)GO TO 8800
      IF(A1.GE.A2.AND.A2.LE.A3.AND.ABS(A2-EXTR).GE.CR)GO TO 8850
      GO TO 8900
 8800 WRITE(6,9004)STIME(I),A2
      EXTR=A2
      GO TO 8900
 8850 WRITE(6,9005)STIME(I),A2
      EXTR=A2
 8900 CONTINUE
 9001 FORMAT(1X,////,10X,18HFIRST EPOCH IS AT ,I5,3X,
     *        10HWITH TIME ,F10.4,5H AND ,
     *        11HWITH VALUE ,F10.4)
 9002 FORMAT(1X,/,10X,18HSECOND EPOCH IS AT  ,I5,3X,
     *        10HWITH TIME ,F10.4,5H AND ,
     *        11HWITH VALUE ,F10.4)
 9003 FORMAT(1H1,///,20X,54(1H*),/,20X,1H*,52X,1H*,/,20X,1H*,17X,
     *        19HDETECTION OF EPOCHS,16X,1H*,/,20X,1H*,52X,1H*,
     *        /,20X,54(1H*))
 9004 FORMAT(20X,F10.4,10X,F11.0)
 9005 FORMAT(20X,F10.4,30X,F11.0)
 9006 FORMAT(1X,////,20X,54(1H*),/,20X,1H*,52X,1H*,/,20X,1H*,13X,
     * 27HLOCAL EXTREMA IN DERIVATIVE,12X,1H*,/,20X,1H*,52X,1H*,/,
     * 20X,54(1H*),////,25X,4HTIME,14X,7HMAXIMUM,13X,7HMINIMUM,/)
 9000 RETURN
      END
```

PROGRAM LISTING (Continued)

```
      SUBROUTINE ROWSRT(A,B,R,N)
C-----------------------------------------------------------------------
C
C * SUBROUTINE ROWSRT SORTS ROWS OF A MATRIX A ACCORDING TO DESCENDIN(
C     ORDER OF MATRIX B.  STORE SORTED A INTO R.
C
C-----------------------------------------------------------------------
      DIMENSION A(810), B(810), R(1620)
      COMMON /SRTCOM/ I,I2,ISORT,RSAVE,SAVER,IR,IA
      LEVEL 2, A,B,R,I,N,I2,ISORT,RSAVE,SAVER,IR,IA
C
C * MOVE SORTING KEY VECTOR TO FIRST COLUMN OF OUTPUT MATRIX AND
C     BUILD ORIGINAL SEQUENCE LIST IN SECOND COLUMN
C
      DO 1000 I = 1,N
        R(I) = B(I)
        I2 = I + N
        R(I2) = I
 1000 CONTINUE
C
C * SORT ELEMENTS IN SORTING KEY VECTOR (ORIGINAL SEQUENCE LIST IS
C     RESEQUENCED ACCORDINGLY)
C
 2000 CONTINUE
      ISORT = 0
      DO 4000 I = 2,N
        IF (R(I) - R(I-1)) 4000, 4000, 3000
 3000   CONTINUE
        ISORT = ISORT + 1
        RSAVE = R(I)
        R(I) = R(I-1)
        R(I-1) = RSAVE
        I2 = I + N
        SAVER = R(I2)
        R(I2) = R(I2-1)
        R(I2-1) = SAVER
 4000 CONTINUE
      IF (ISORT) 2000, 5000, 2000
C
C * MOVE ROWS FROM MATRIX A TO MATRIX R (NUMBER IN SECOND COLUMN OF R
C     REPRESENTS ROW NUMBER OF MATRIX A TO BE MOVED)
C
 5000 CONTINUE
      DO 8000 I = 1,N
C
C * MOVE ELEMENT TO OUTPUT MATRIX
C
        I2 = I + N
        R(I) = A( R(I2) )
        R(I2) = B( R(I2) )
 8000 CONTINUE
      RETURN
      END
```

PROGRAM LISTING (Continued)

```
      OVERLAY(P,11,0)
      PROGRAM REG2AN
C-----------------------------------------------------------------------
C
C * SUBROUTINE REG2AN --SPECIAL ANALYSIS OF REGIME II
C
C-----------------------------------------------------------------------
      COMMON / SSDATA / NSSPTS , SDTIME , SUBSTR(5000) , SSTIME(5000)
      COMMON / REGII  / MAXPT , MINPT , VALMAX , VALMIN
      LEVEL 2, NSSPTS,SDTIME,SUBSTR,SSTIME,MAXPT,MINPT,VALMAX,VALMIN
C
C * CALCULATE MAXIMUM AND MINIMUM VALUES
C
      VALMAX = -99999.
      VALMIN =  99999.
      DO 4000 I = 1,NSSPTS
        IF (SUBSTR(I) .GT. VALMAX) GO TO 1000
        IF (SUBSTR(I) .LT. VALMIN) GO TO 2000
        GO TO 4000
 1000 CONTINUE
C
C * SUBSTR(I) IS NEW MAXIMUM
C
        VALMAX = SUBSTR(I)
        MAXPT = I
        GO TO 4000
 2000 CONTINUE
C
C * SUBSTR(I) IS NEW MINIMUM
C
        VALMIN = SUBSTR(I)
        MINPT = I
 4000 CONTINUE
C
C
C * RETURN  MAXPT  = INDEX OF MAXIMUM
C           MINPT  = INDEX OF MINIMUM
C           VALMAX = MAXIMUM
C           VALMIN = MINIMUM
C
 9000 CONTINUE
      RETURN
      END
```

PROGRAM LISTING (Continued)

APPENDIX A.2

IMSL ROUTINES IN IMSL2 LIBRARY

APPENDIX A.2

IMSL ROUTINES IN IMSL2 LIBRARY

CALLED BY MOOSE	CALLED BY ANOTHER IMSL ROUTINE
RLFOR	RLFOTW
GFIT	RLDCW
FTFREQ	RLPOLY
MDNOR	RLOPDC
	MDBETI
	MDBETA
	MLGAMA
	RLPRDI
	RLDCVA
	RLDOPM
	FTAUTO
	MDCH
	MGAMMA
	VERTST

APPENDIX B

PROGRAM INPUT

APPENDIX B.1

JOB CONTROL LANGUAGE FOR RUNNING THE PROGRAM

APPENDIX B.1

JOB CONTROL LANGUAGE FOR RUNNING THE PROGRAM

Job Control Language for running MOOSE UPDATE file with Level 2 IMSL and CALCOMP Plotting Routines.

```
Job name,T1ØØ,STMFZ.
Account card
FILE,TAPE1,RT=W.
FILE,OLDPL,RT=S.
REQUEST,TAPE1Ø,*PF.
ATTACH,IMSL2L,ID =   .
LIBRARY,*,IMSL2L.
BEGIN,ATTACH,PLOTLIB.
COPYCR,INPUT,DEBG.
REWIND,DEBG.
GETPF,OLDPL,MOOSEOLDPL,ID =    ,ST=MFA.
UPDATE,F,W,L=1.
FTN,I,B=BMOOSE,L=Ø,EL=F,D=DEBG,PL=65ØØ.
ATTACH,TAPE1,inductance coil data file name,ID =   .
MAP,OFF.
BMOOSE.
BEGIN,PLOT,CALCOMP,TAPE1Ø.
EXIT.
BEGIN,PLOT,CALCOMP,TAPE1Ø.
EXIT.
*EOR
CbbbbbDEBUG
(Optional user debug directives)
*EOR
*IDENT TEMP
*EOR
     user input data deck
*EOR
*EOF
```

APPENDIX B.2

SUPPLEMENTAL UPDATE IDENT FOR
OBTAINING PAGE SIZE PLOTS

APPENDIX B.2

SUPPLEMENTAL UPDATE IDENT FOR OBTAINING PAGE SIZE PLOTS

```
*IDENT PLT
*D PR-SCRIB4.29,4Ø
      XLNGTH=11.
      YLNGTH=8.5
      XSIZE=8.Ø
      YSIZE1=4.
      YSIZE2=1.5
      XBASE=2.
      YBASE=3.Ø
      CALL FIXSCA(XARRAY(1),NDPTS,XSIZE,XSCALE,XOR,XMAX,DX)
      CALL FIXSCA(YARRY1(1),NDPTS,YSIZE1,YSCALE,YOR,YMAX,DY)
*D PR-SCRIB4.5Ø
      CALL LABELA(Ø.,DY,XOR,XMAX,YOR,YMAX,ONE,ONE)
*D PR-SCRIB4.54,6Ø
      CALL FIXSCA(YARRY2(1),NDPTS,YSIZE2,YSCALE,YOR,YMAX,DY)
      YBASE=1.
*D PR-SCRIB4.66
      CALL PLTAXS(DX,DY,XOR,XMAX,YOR,YMAX,MODE)
*D PR-SCRIB4.86
      YMIN1=YOR+(YSCALE*6.3)
*D PR-SCRIB4.89
      CALL PLTSYM(Ø.15,ISENT,Ø.,XOR,YMIN1)
*D PR-SCRIBL.28,4Ø
      XLNGTH=11.
      YLNGTH=8.5
      XSIZE=8.Ø
      YSIZE=6.Ø
      XBASE=2.Ø
      YBASE=1.Ø
      CALL FIXSCA(XARRAY(1),NDPTS,XSIZE,XSCALE,XOR,XMAX,DX)
      CALL FIXSCA(YARRY1(1),NDPTS,YSIZE,YSCALE,YOR,YMAX,DY)
*D PR-SCRIBL.66,69
```

SUPPLEMENTAL UPDATE IDENT FOR OBTAINING PAGE SIZE PLOTS (cont.)

```
*D PR-SCRIBL.71
      YMAX1=YMAX+1.Ø*YSCALE
*D PR-SCRIBL.74
      CALL PLTSYM(Ø.15,ISENT,Ø.,XOR,YMAX1)
*D PR-SCRIB1.25,37
      XLNGTH=11.
      YLNGTH=8.5
      XSIZE=8.Ø
      YSIZE=6.Ø
      XBASE=2.Ø
      YBASE=1.Ø
      CALL FIXSCA(XARRAY(1),NDPTS,XSIZE,XSCALE,XOR,XMAX,DX)
      CALL FIXSCA(YARRAY(1),NDPTS,YSIZE,YSCALE,YOR,YMAX,DY)
*D PR-SCRIB1.56,59
*D PR-SCRIB1.61
      YMAX=YMAX+1.Ø*YSCALE
*D PR-SCRIB1.64
      CALL PLTSYM(Ø.15,ISENT,Ø.,XOR,YMAX1)
*D PR-PAWS2.256,268
      XLNGTH=11.
      YLNGTH=8.5
      XSIZE=8.Ø
      YSIZE=6.Ø
      XBASE=2.Ø
      YBASE=1.Ø
      CALL FIXSCA(XARRAY(1),NDPTS,XSIZE,XSCALE,XOR,XMAX,DX)
      CALL FIXSCA(YARRAY(1),NDPTS,YZISE,YSCALE,YOR,YMAX,DY)
*D PR-PAWS2.287,29Ø
*D PR-PAWS2.292
      YMAX1=YMAX+1.Ø*YSCALE
*D PR-PAWS2.295
      CALL PLTSYM(Ø.15,ISENT,Ø.,XOR,YMAX1)
```

APPENDIX B.3

INPUT DATA DESCRIPTION

APPENDIX B.3

INPUT DATA DESCRIPTION

Input Level	Subroutine/Overlay	Card	Format	Variable	Description
1	IAINPT (A,0,0)				Called only once.
		1	"...","..."	USRNAM(2)	User name, 2 names each with a maximum of 10 characters.
2	RDPURP (A,0,0)				Called only once.
		1	I5	NLINES	Number of cards used to enter purpose of run.
		2	6A10	PURPOS(6)	Purpose of run -line 1
		NLINES+1			Purpose of run -line NLINE
3	ANALIZ (A,0,0)				Repetitively used to control analysis.
		1	I2	IANOPT	Analysis option =0 No further analysis. Program terminates. =1 User controlled analysis. Control passes to subroutine KITTY8. =2 Complete data record analysis. Control passes to subroutine COMPLT.

**** END OF INPUT DECK IF IANOPT = 0 ****

**** NEXT CARD IS INPUT LEVEL 4 OF ****

**** KITTY8 OR COMPLT INPUT OPTIONS ****

Input Level	Subroutine/Overlay	Card	Format	Variable	Description
4	KITTY8 (A,0,0)				Optionally called by ANALIZ IANOPT = 1
		1	I2	SUBOPT	Analysis option =0 No further analysis. Control returns to ANALIZ. =1 Select data subset. Control passes to subroutine SUBSET. =2 Power spectrum. Control passes to subroutine PAWS. SUBOPT=1 must be called before this option. =3 Forward-backward fits. Control passes to subroutine GIRAFE. Not currently available. =4 Polynomial and periodic fits, noise analysis. Control passes to subroutine THEFFT. SUBOPT=1 must be called before this option. SUBOPT=11 must be called before this option. =5 Smoothing. Control passes to subroutine SMOOTH. SUBOPT=1 must be called before this option. =11 First and second derivatives of entire data set. Control passes to subroutine DERIV. =12 Detection of two epochs. Control passes to subroutine DETECT. SUBOPT=11 must be called before this option.

Input Level	Subroutine/Overlay	Card	Format	Variable	Description
	**** END OF KITTY8 INPUT IF SUBOPT = 0 ****				
	**** NEXT CARD IS INPUT LEVEL 5 ****				
	**** SEE APPROPRIATE SUBROUTINE BELOW ****				
5	SUBSET (C,2,0)				Optionally called by KITTY8 with SUBOPT=1
	SUBPAR (C,2,0)	1	F10.4	TINIT	Initial time of data subset, milliseconds.
			F10.4	TLAST	Final time of data subset, milliseconds.
			I10	NODTS	Number of 2.5 micro-second intervals between points in data subset.
5	PAWS (D,3,0)				Optionally called by KITTY8 with SUBOPT=2
		1	I2	PLOTOP	Plotting option for power spectrum. =0 Plot off. =1 Plot on.
			I2	LAGOPT	Frequency option. =0 Frequencies resolution is maximum permitted, multiples of .25, .5, 1.0 or 2.0 kHz. =1 Frequencies are multiples of 1 kHz.
5	GIRAFE (E,4,0)				Optionally called by KITTY8 with SUBOPT=3 Not currently available. No input required.
5	THEFFT (F,5,0)				Optionally called by KITTY8 with SUBOPT=4
		1	I2	PLOTOP(1)	Plotting option for data fits. =0 Plot off. =1 Plot on.

INPUT DATA DESCRIPTION (cont.)

Input Level	Subroutine/Overlay	Card	Format	Variable	Description
			I2	PLOTOP(2)	Plotting option for entire fitted data set. =0 Plot off. =1 Plot on.
	PAWS (I,5,2)				Called by THEFFT
		2	I2	PLOTOP	Plotting option for power spectrum. =0 Plot off. =1 Plot on.
5	SMOOTH (G,6,0)				Optionally called by KITTY8 with SUBOPT=5
		1	I2	SMOPT	Smoothing option. =1 11-point local median. =2 11-point local mean.
5	DERIV (L,7,0)				Optionally called by KITTY8 with SUBOPT=11
		1	I2	PLOTOP	Plotting option for first derivatives. =0 Plot off. =1 Plot on.
5	DETECT (N,10,0)				Optionally called by KITTY8 with SUBOPT=12 No input required.

**** NEXT CARD IS INPUT LEVEL 4 FOR NEXT ****

**** ANALYSIS OPTION OF SUBROUTINE KITTY8 ****

Input Level	Subroutine/Overlay	Card	Format	Variable	Description
4	COMPLT (A,0,0)				Optionally called by ANALIZ IANOPT = 2
	DERIV (L,7,0)	1	I2	PLOTOP	Plotting option for derivatives. =0 Plot off. =1 Plot on.

INPUT DATA DESCRIPTION (cont.)

Input Level	Subroutine/Overlay	Card	Format	Variable	Description
	COMPLT (A,0,0)	2	I2	IDIV	Option for division of Regime I =-1 Divide Regime I =0 Do not divide Regime I =1 No further analysis Control returns to ANALIZ

**** NEXT CARD IS INPUT LEVEL 3 IF IDIV=1 ****

**** NEXT CARD IS CARD 4 IF IDIV=0 ****

**** NEXT CARD IS CARD 3 IF IDIV=-1 ****

Input Level	Subroutine/Overlay	Card	Format	Variable	Description
	COMPLT (A,0,0)	3	F8.4	SUBTIM	Time of subdivision of Regime I in milliseconds
	THEFFT (F,5,0)	4	I2	PLOTOP(1)	Plotting option for Regime I data fits. =0 Plot off. =1 Plot on.
			I2	PLOTOP(2)	=0 Required input.
	PAWS (I,5,2)	5	I2	PLOTOP	Plotting option for Regime I power spectrum. =0 Plot off. =1 Plot on.
	PAWS (D,3,0)	6	I2	PLOTOP	Plotting option for Regime I noise power spectrum. =0 Plot off. =1 Plot on.
			I2	LAGOPT	Frequency option for Regime I noise. =0 Frequencies resolution is minimum permitted, multiples of .25, .5, 1.0 or 2.0 kHz =1 Frequencies are multiples of 1.0 kHz.

**** REPEAT CARDS 4, 5, AND 6 FOR ****

**** PART 2 OF REGIME I IF IDIV=-1 ****

INPUT DATA DESCRIPTION (cont.)

Input Level	Subroutine/Overlay	Card	Format	Variable	Description
	PAWS (D,3,0)	7	I2	PLOTOP	Plotting option for Regime II power spectrum. =0 Plot off. =1 Plot on.
			I2	LAGOPT	=1 Required input.
	THEFFT (F,5,0)	8	I2	PLOTOP(1)	Plotting option for Regime III data fits. =0 Plot off. =1 Plot on.
			I2	PLOTOP(2)	Plotting option for entire fitted data set. =0 Plot off. =1 Plot on.
	PAWS (I,5,2)	9	I2	PLOTOP	Plotting option for Regime III power spectrum. =0 Plot off. =1 Plot on.
	PAWS (D,3,0)	10	I2	PLOTOP	Plotting option for Regime III noise power spectrum. =0 Plot off. =1 Plot on.
			I2	LAGOPT	Frequency option for Regime III noise. =0 Frequencies resolution is maximum permitted, multiples of .25, .5, 1.0 or 2.0 kHz. =1 Frequencies are multiples of 1.0 kHz.

**** NEXT CARD IS INPUT LEVEL 3 ****

**** ANALYSIS OPTION OF SUBROUTINE ANALIZ ****

APPENDIX B.4

SAMPLE INPUT DATA

APPENDIX B.4.

SAMPLE INPUT DATA

This data deck will produce the output found in Appendix C.2, Sample Output.

```
*EOR
"SAMPLE"
    1
 TEST - CHECK COMPLT LEG
 2
 1
 Ø
 1 Ø
 1
 1 Ø
 1 1
 1 1
 1
 1 Ø
*EOR
```

APPENDIX C

PROGRAM OUTPUT

APPENDIX C.1

OUTPUT DATA DESCRIPTION

APPENDIX C.1

OUTPUT DATA DESCRIPTION

User's Name

Purpose of Run

Analysis Option

Data Record Description

- start time
- number of point
- point interval

Data Record Classification

- saturated points
- classification

Portions of a Complete Analysis Follows, KITTY8 User's Option will vary

Derivative

- no printout

Epoch Detection

- index, time, values of Epochs 1 and 2

Optional Division of Regime I

Analysis of Regime I, III, or Parts of Regime I

- Regime Fitting and Analysis (THEFFT Subroutine)
  - polynomial coefficients for x, x^2, x^3, x^4, x^0
- Power Spectrum (PAWS Routine)
  - mean and variance of waveform
  - lag and autocovariance
  - frequency and power
  - ten predominant frequencies
- Periodic Curve
  - frequency, cosine and sine coefficients
- Fitted Curves
  - index, time, data, polynomial fit, data minus polynomial fit, periodic fit, noise
- Analysis of Noise
  - mean, standard deviation
  - χ^2 test, χ^2 and Q statistics

OUTPUT DATA DESCRIPTION (cont.)

Shock Detection

for points at least 2.5 standard deviations from the mean: index, time, noise value, standard deviations from mean, a '*' if more than 3., standard deviations from mean of derivative

Analysis of Regime II

maximum and minimum

APPENDIX C.2

SAMPLE OUTPUT

APPENDIX C.2

SAMPLE OUTPUT

The following plots were produced using the input data listed in Appendix B.4, and the UPDATE deck for obtaining page size plots, in Appendix B.2.

1. Top: Plot of data as stored on tape.
   Bottom: Plot of first derivative of data.

2. Power spectrum of Regime I.

3. Solid line: Data on Regime I.
   Broken line: 4^{th} degree polynomial fit to data in Regime I.

4. Solid line: Regime I data after subtraction of polynomial.
   Broken line: Fourier series fit using ten predominant frequencies.

5. Regime I noise; data with polynomial and Fourier fit subtracted.

6. Power spectrum of Regime I data after subtraction of polynomial.

7. Power spectrum of Regime II.

8. Power spectrum of Regime III.

9. Solid line: Data in Regime III.
   Broken line: 4^{th} degree polynomial fit to data in Regime III.

10. Solid line: Regime III data after subtraction of polynomial.
    Broken line: Fourier series fit using ten predominant frequencies.

11. The Regime III noise; data after subtraction of polynomial and Fourier fit.

12. Entire data set with noise from Regimes I and III removed.

13. Power spectrum of Regime III after subtraction of polynomial.

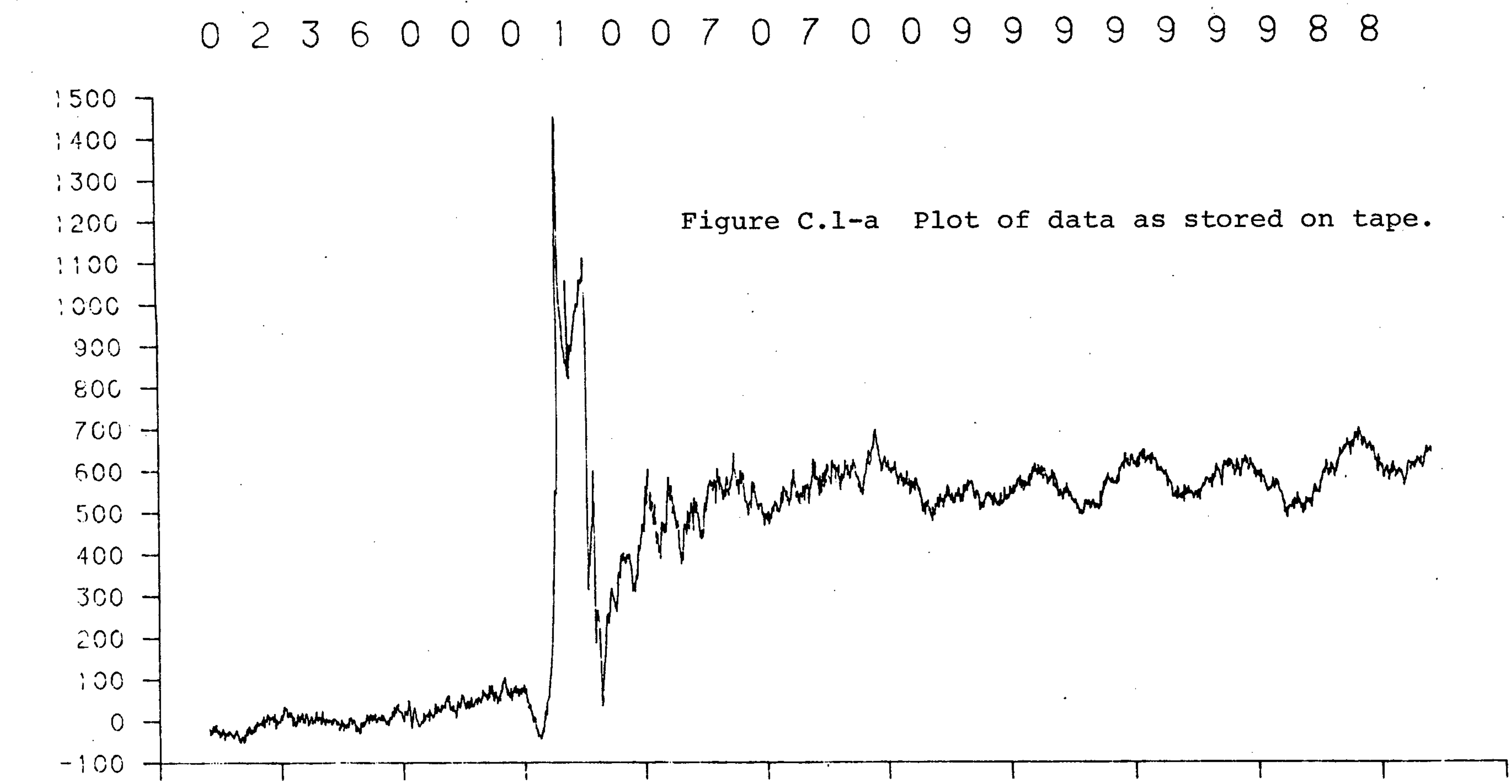


Figure C.1-a Plot of data as stored on tape.

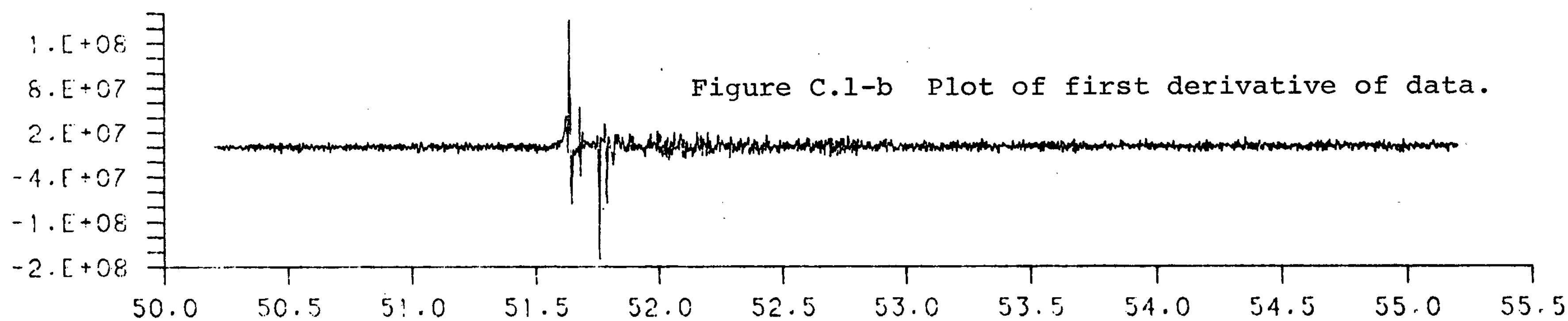


Figure C.1-b Plot of first derivative of data.

0 2 3 6 0 0 0 1 0 0 7 0 7 0 0 9 9 9 9 9 9 9 8 8

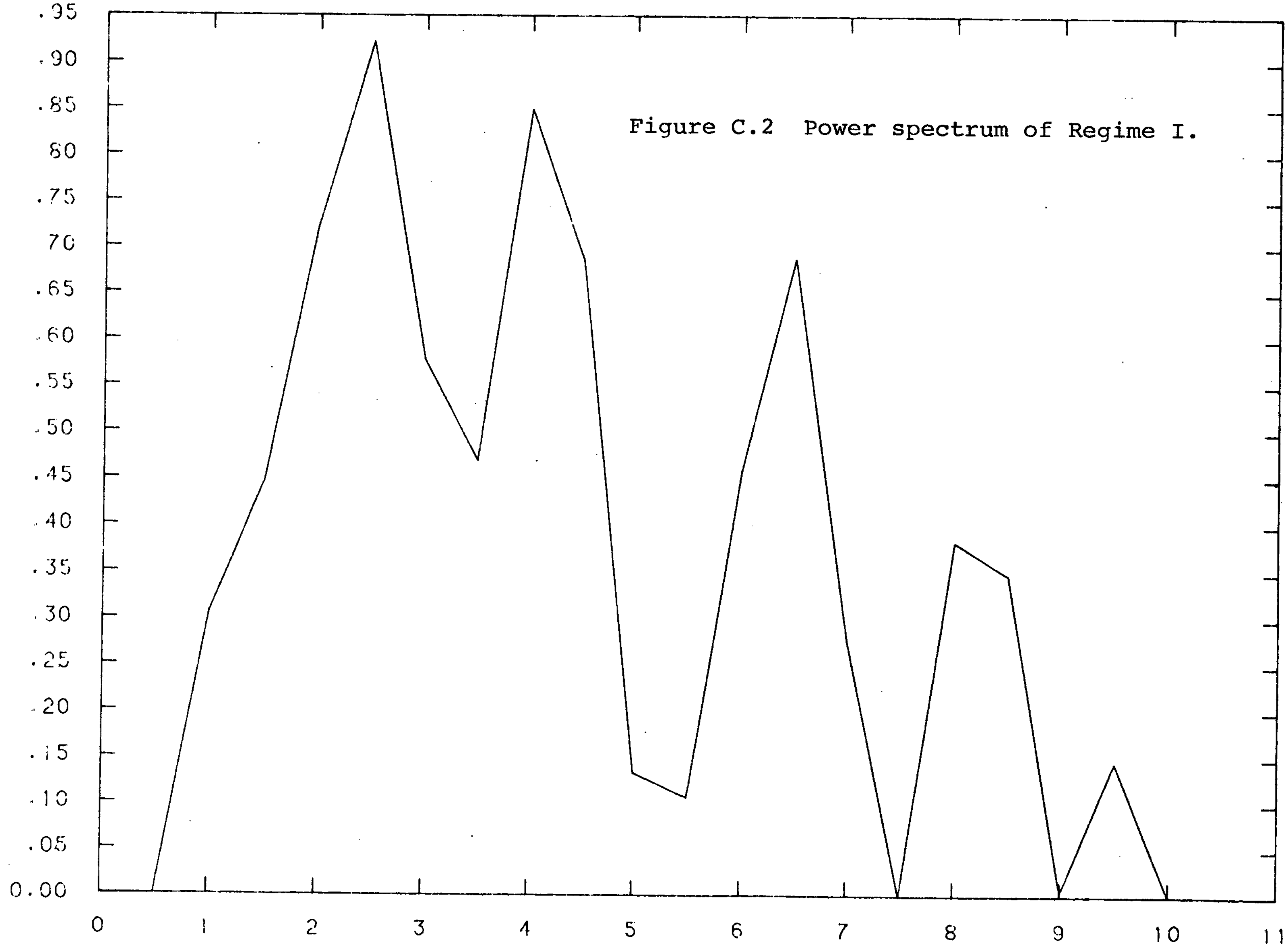


Figure C.2 Power spectrum of Regime I.

0 2 3 6 0 0 0 1 0 0 7 0 7 0 0 9 9 9 9 9 9 9 8 8

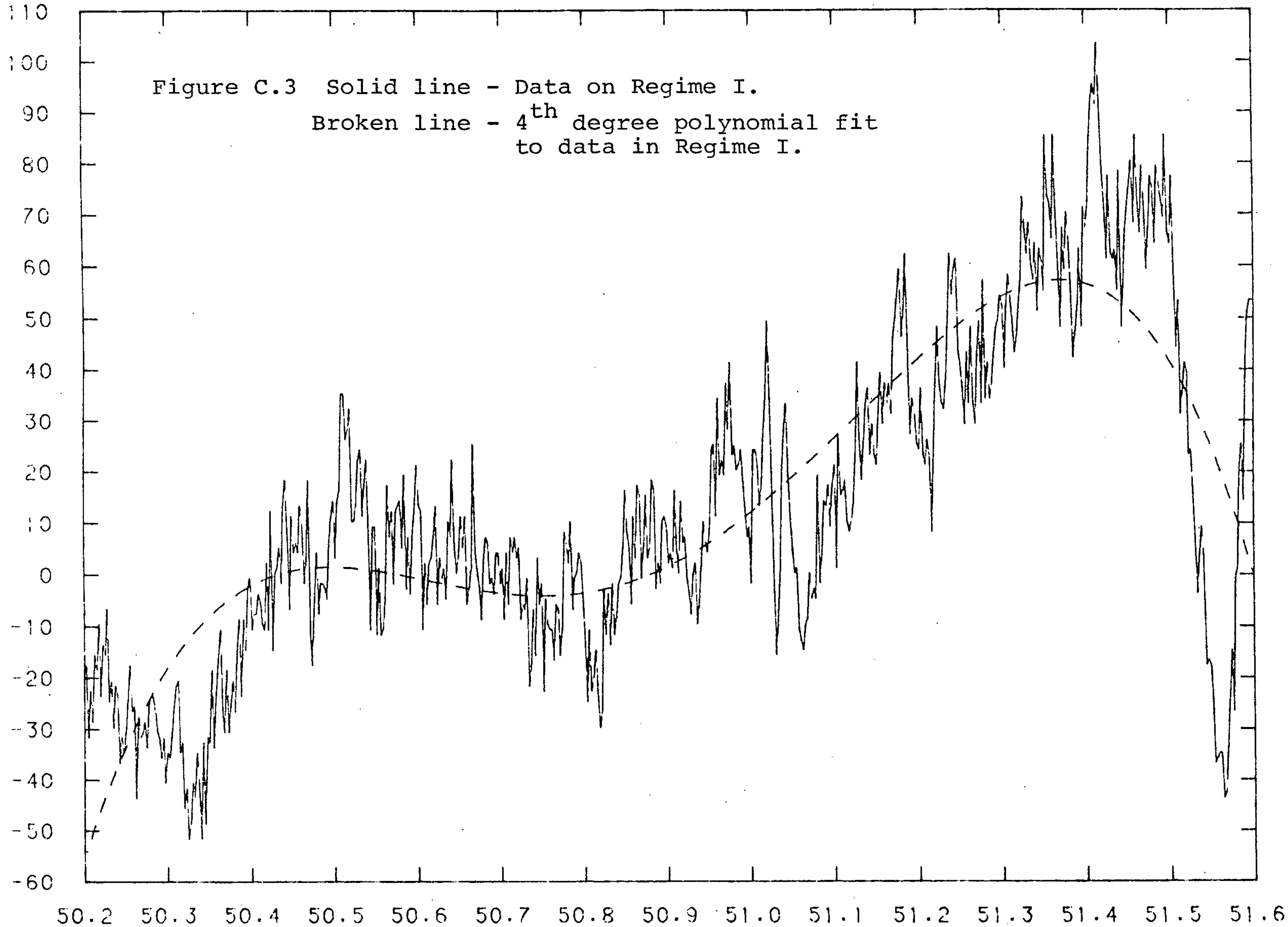


Figure C.3 Solid line - Data on Regime I.
Broken line - 4^{th} degree polynomial fit to data in Regime I.

0 2 3 6 0 0 0 1 0 0 7 0 7 0 0 9 9 9 9 9 9 9 8 8

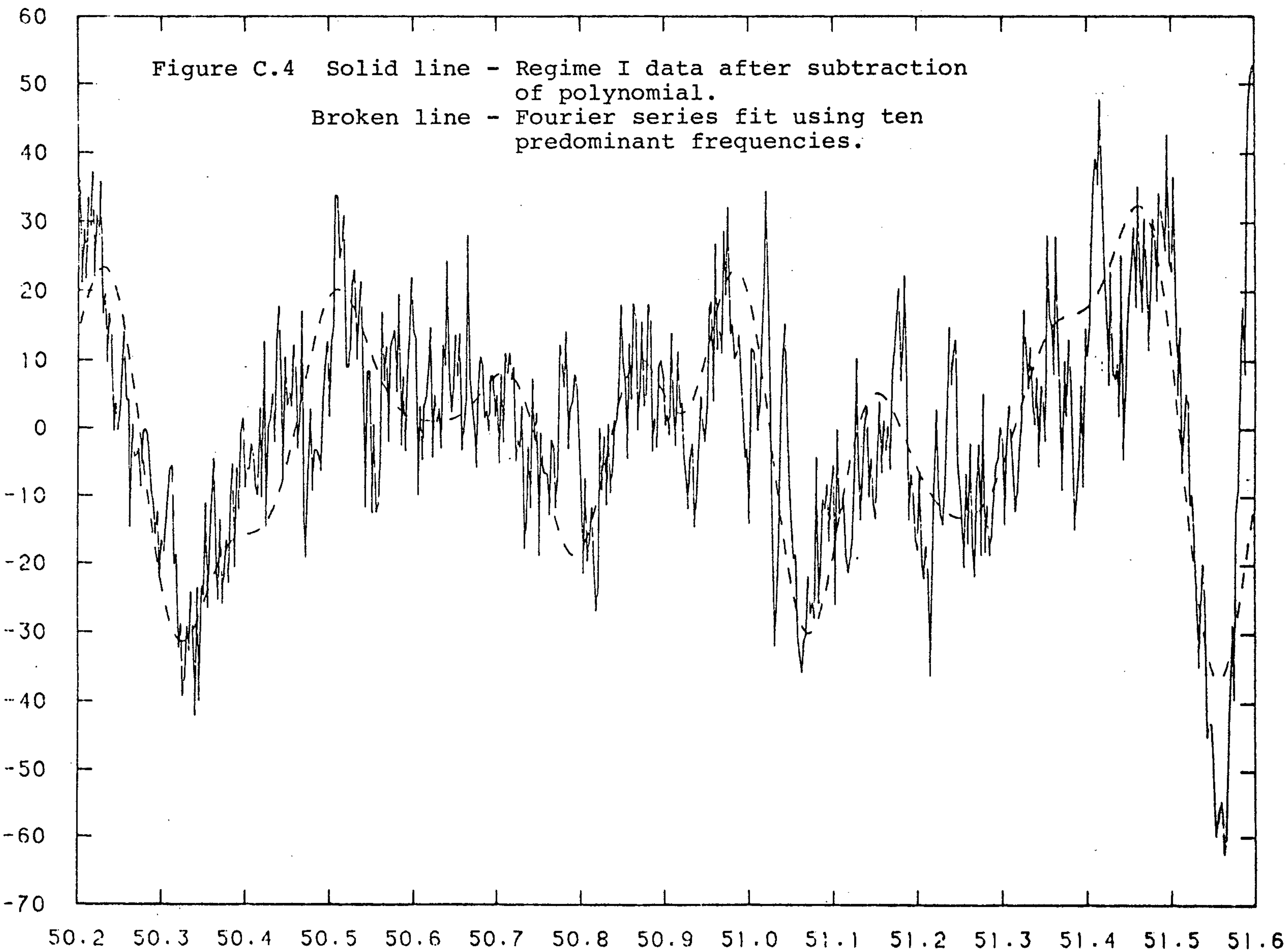


Figure C.4 Solid line - Regime I data after subtraction of polynomial.
Broken line - Fourier series fit using ten predominant frequencies.

0 2 3 6 0 0 0 1 0 0 7 0 7 0 0 9 9 9 9 9 9 9 8 8

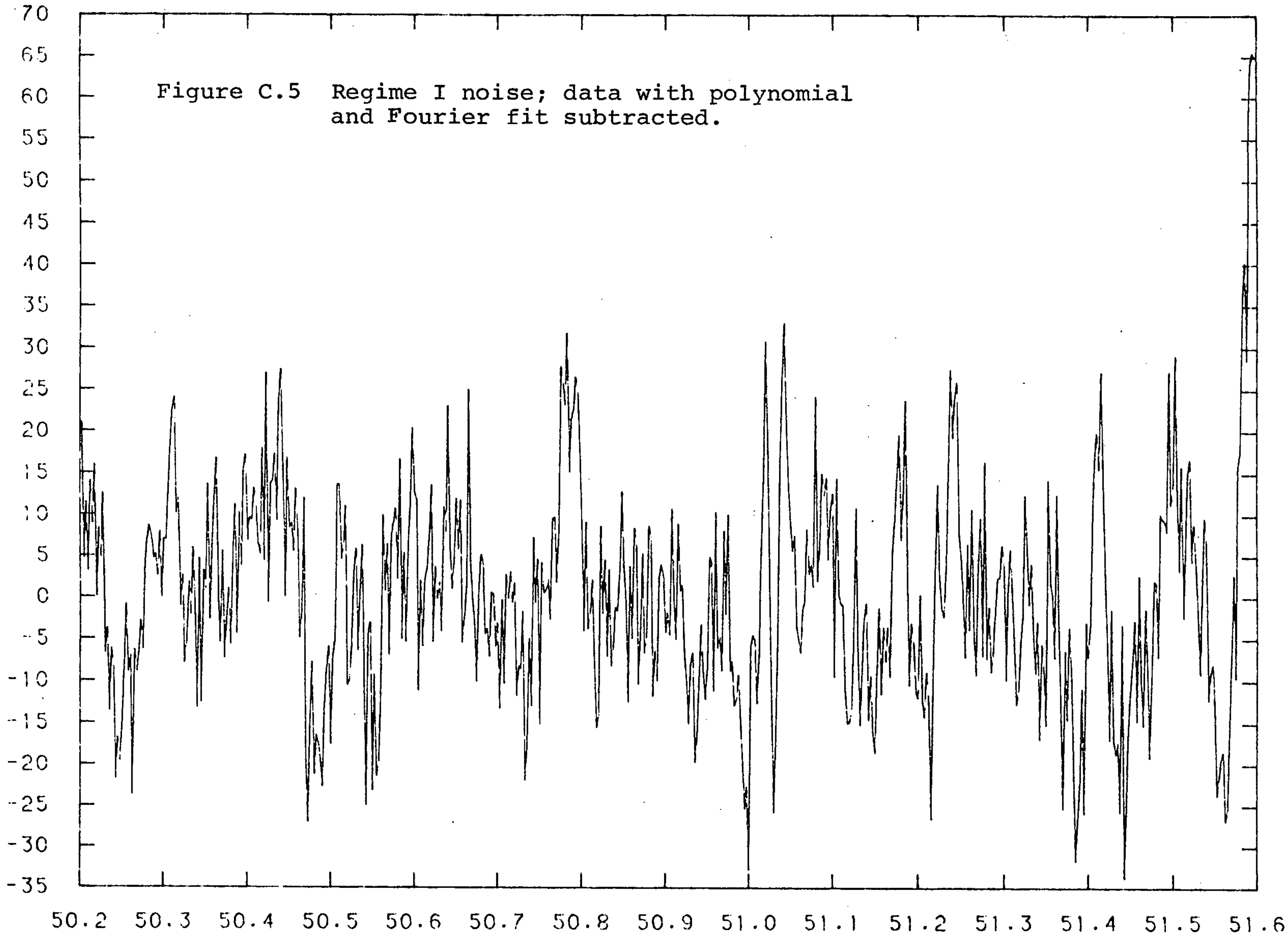


Figure C.5 Regime I noise; data with polynomial and Fourier fit subtracted.

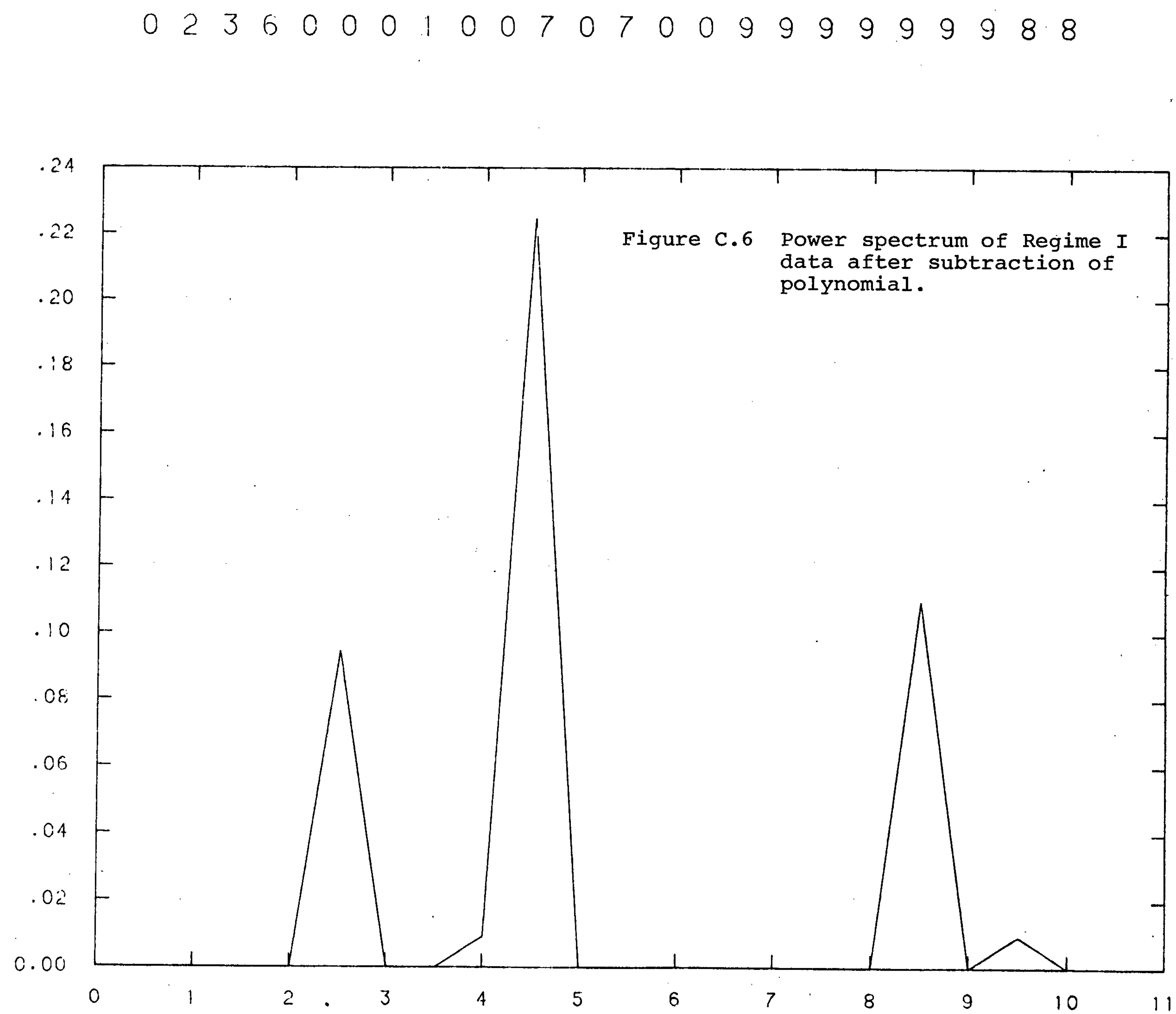


Figure C.6 Power spectrum of Regime I data after subtraction of polynomial.

0 2 3 6 0 0 0 1·0 0 7 0 7 0 0 9 9 9 9 9 9 9 8 8

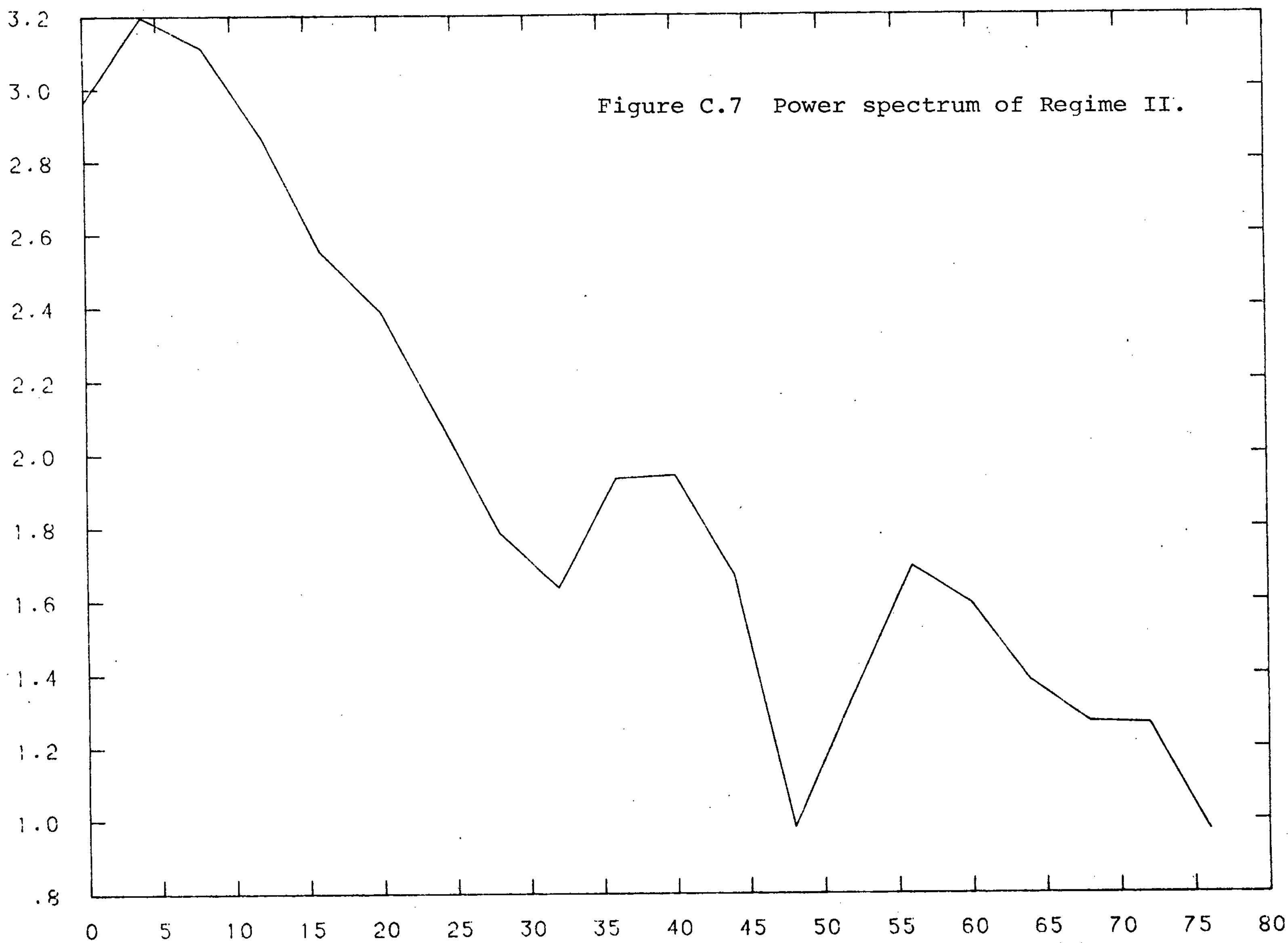


Figure C.7 Power spectrum of Regime II.

0 2 3 6 0 0 0 1 0 0 7 0 7 0 0 9 9 9 9 9 9 9 8 8

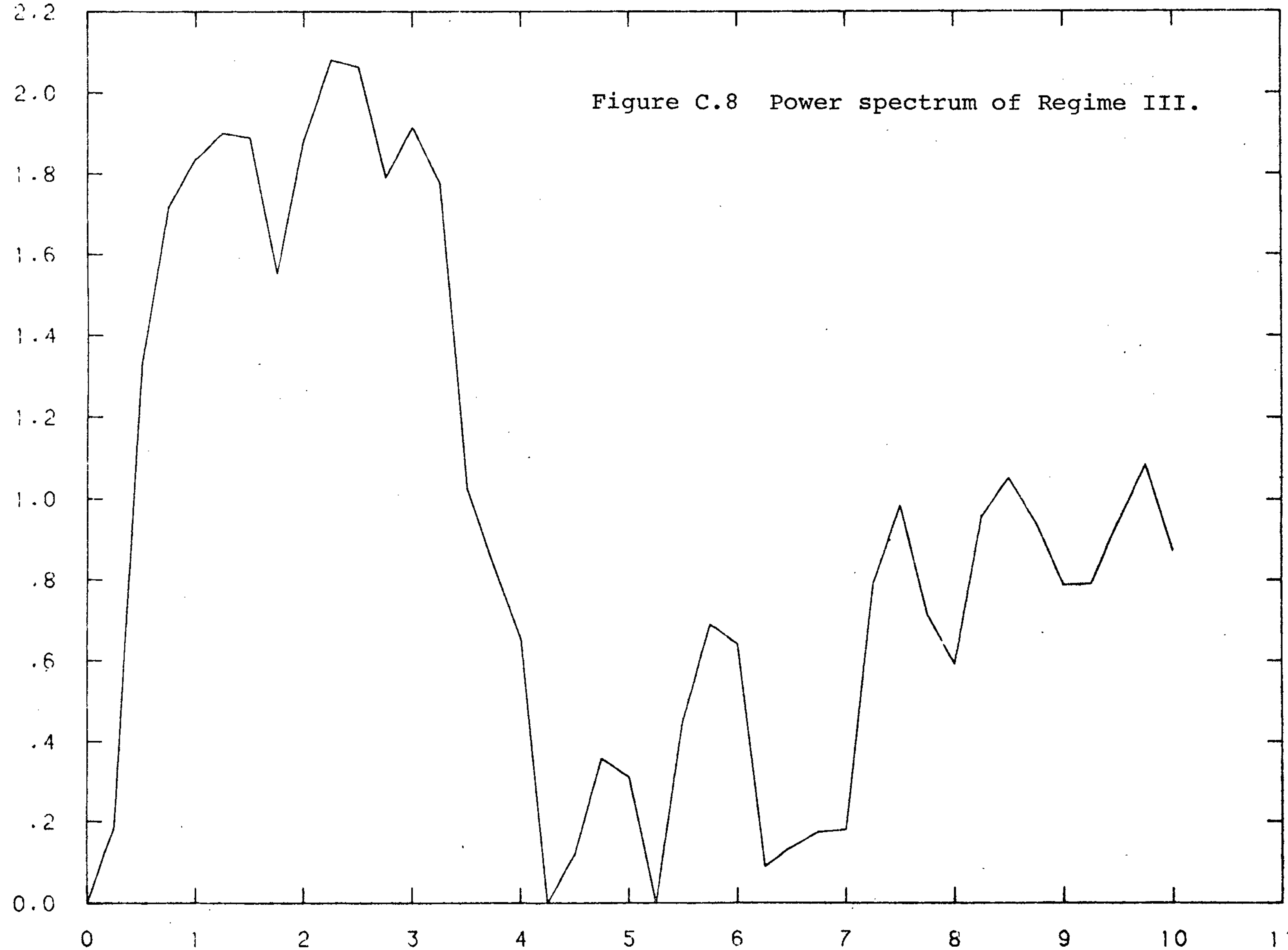


Figure C.8 Power spectrum of Regime III.

0 2 3 6 0 0 0 1 0 0 7 0 7 0 0 9 9 9 9 9 9 9 8 8

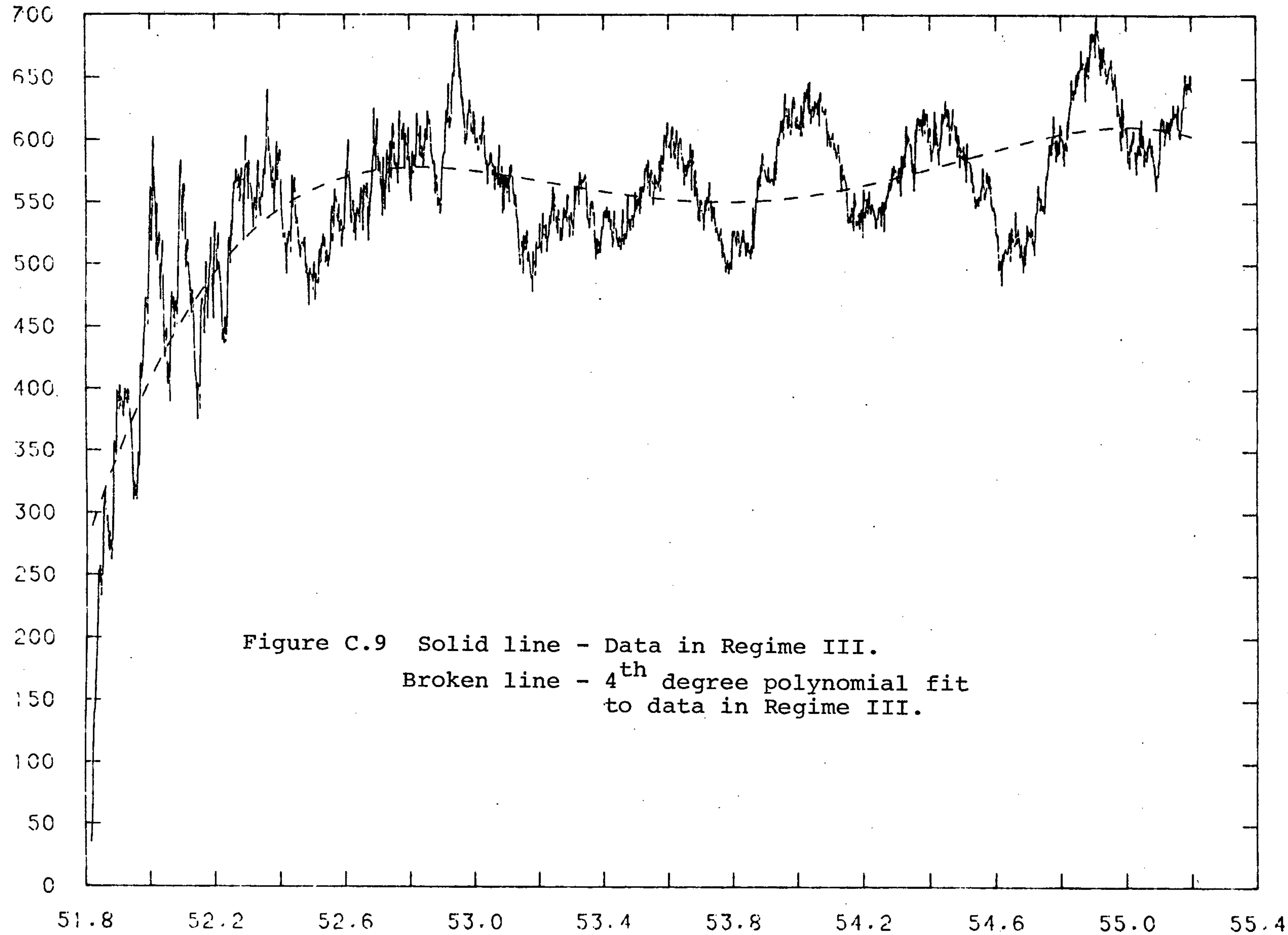


Figure C.9 Solid line - Data in Regime III.
Broken line - 4^{th} degree polynomial fit to data in Regime III.

0 2 3 6 0 0 0 1 0 0 7 0 7 0 0 9 9 9 9 9 9 9 8 8

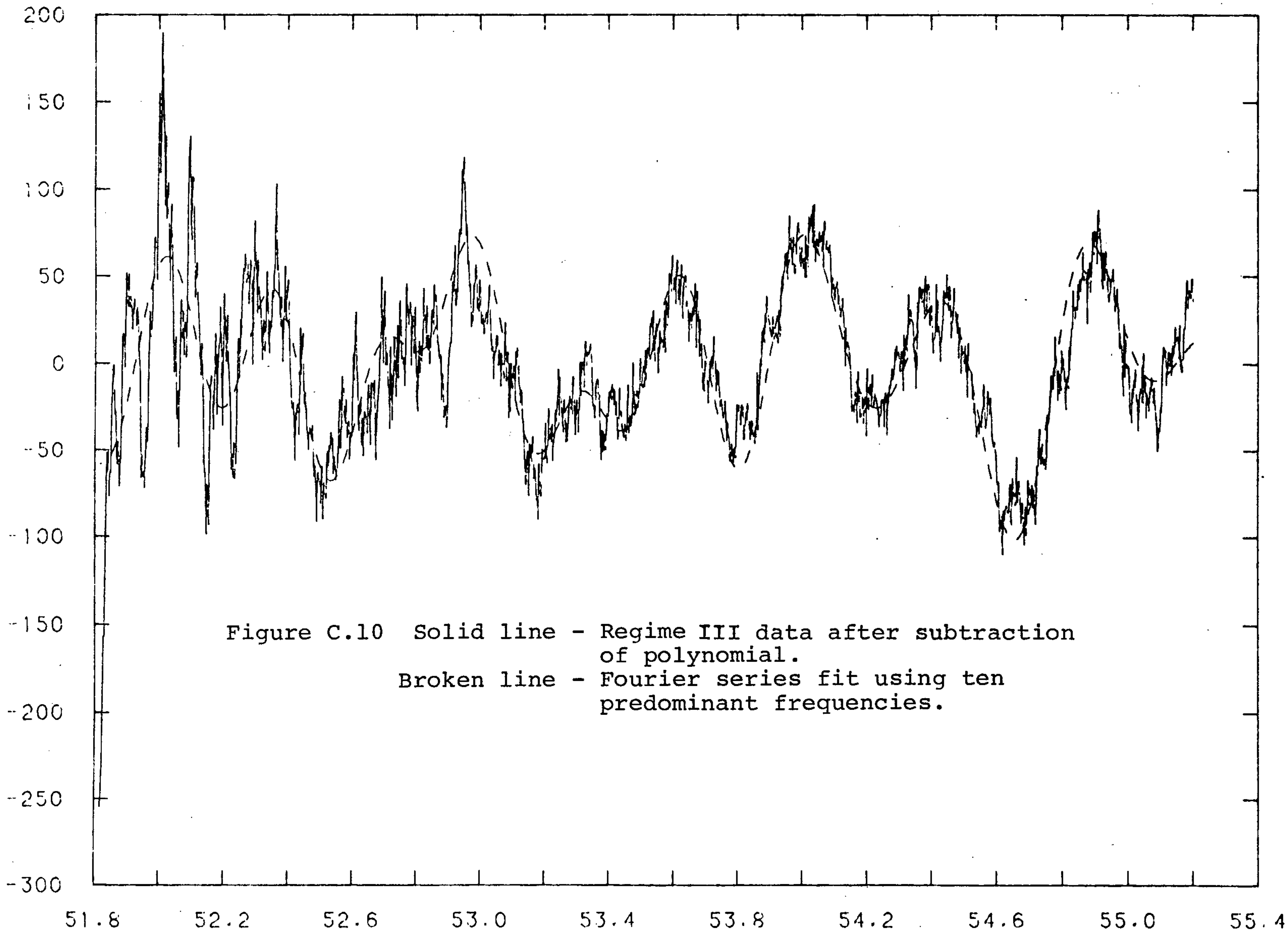


Figure C.10 Solid line - Regime III data after subtraction of polynomial.
Broken line - Fourier series fit using ten predominant frequencies.

0 2 3 6 0 0 0 1 0 0 7 0 7 0 0 9 9 9 9 9 9 9 8 8

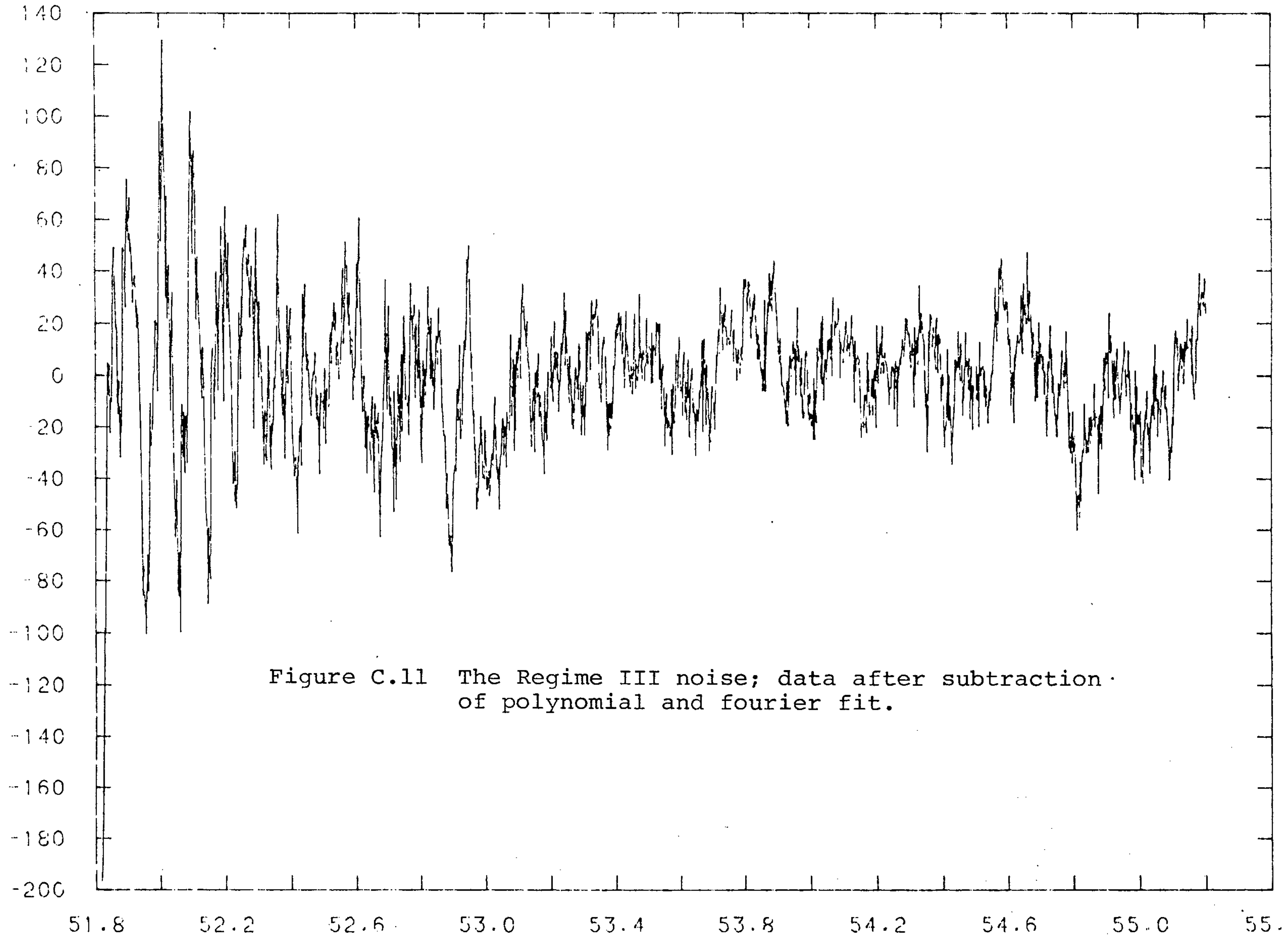


Figure C.11 The Regime III noise; data after subtraction of polynomial and fourier fit.

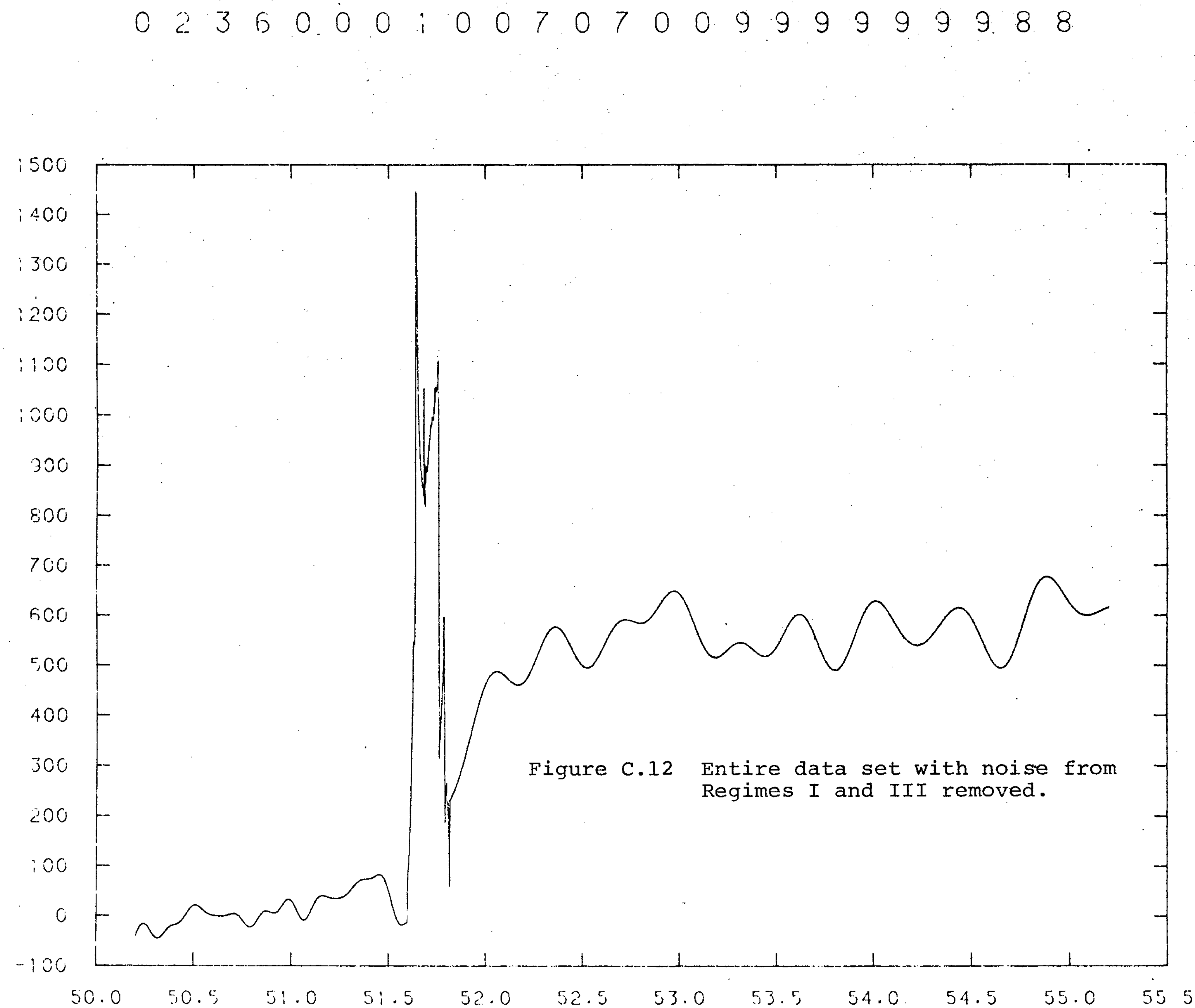


Figure C.12 Entire data set with noise from Regimes I and III removed.

0 2 3 6 0 0 0 1 0 0 7 0 7 0 0 9 9 9 9 9 9 9 8 8

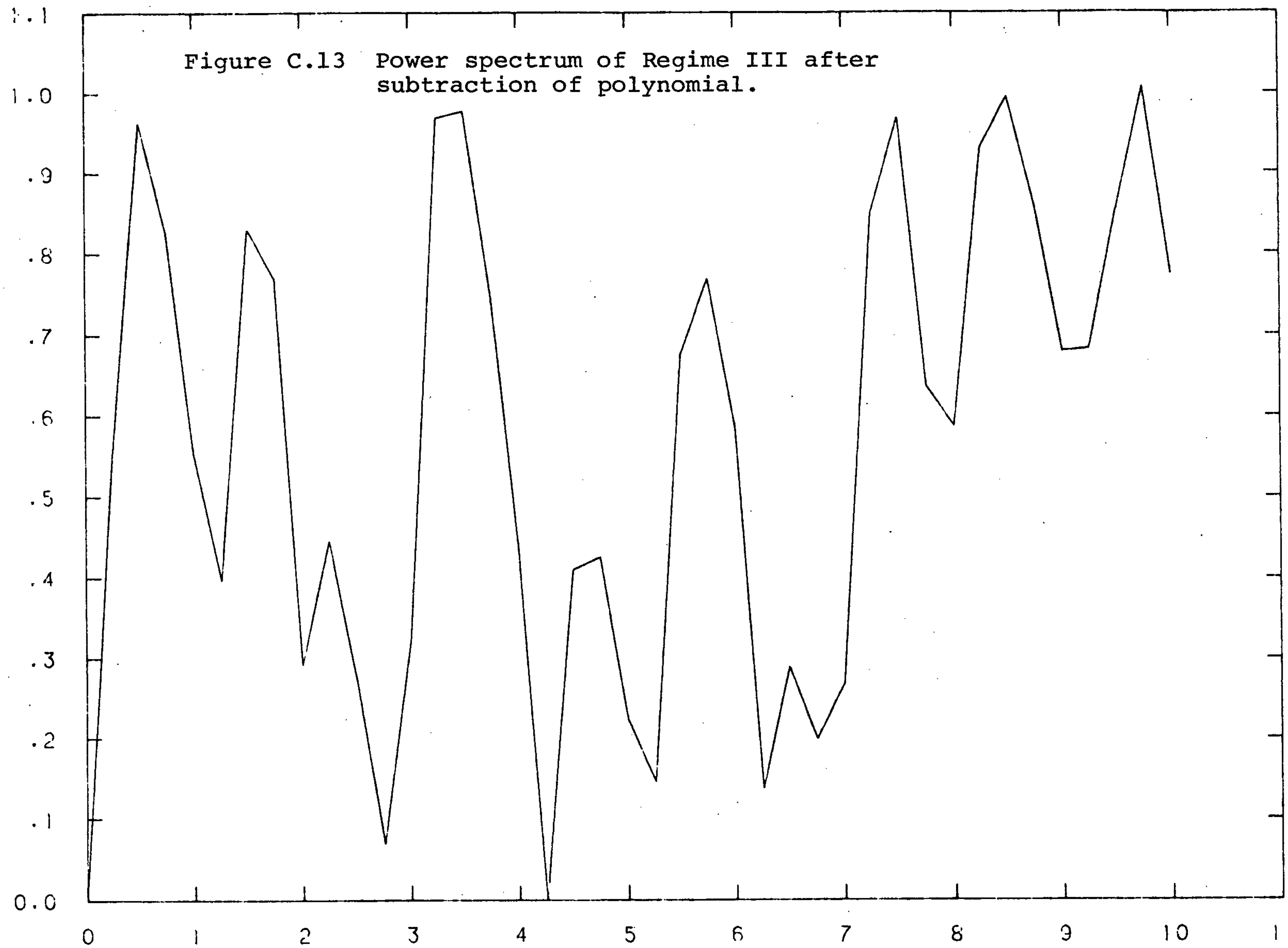


Figure C.13 Power spectrum of Regime III after subtraction of polynomial.

APPENDIX C.3

EXECUTION ERROR CODE DESCRIPTION

APPENDIX C.3

EXECUTION ERROR CODE DESCRIPTION

PERROR ERROR CODE	SUBROUTINE/OVERLAY	ERROR DESCRIPTION
5	ANALIZ (A,0,0)	IMPROPER ANALYSIS OPTION. IANOPT < 0
8	ANALIZ (A,0,0)	IMPROPER ANALYSIS OPTION. IANOPT > 2
51	SUBPAR (C,2,0)	IMPROPER INPUT TO SUBROUTINE. TLAST ≤ TINIT, DATA SUBSET AND POINTS
52	SUBPAR (C,2,0)	IMPROPER INPUT TO SUBROUTINE. POINT INTERVAL BETWEEN DATA SUBSET POINTS, NODTS ≤ 0
53	SUBSET (C,2,0)	CALCULATION ERROR. IS ≤ 0, DATA SUBSET STARTING INDEX
54	SUBSET (C,2,0)	CALCULATION ERROR. IE ≤ IS, ENDPOINT INDICES OF DATA SUBSET
55	SUBSET (C,2,0)	DATA SUBSET ERROR. NSSPTS ≤ 0, NUMBER OF SUBSET DATA POINTS
57	PAWS (D,3,0)	ERROR IN INPUT PARAMETER ARRAY TO IMSL SUBROUTINE FTFREQ. IND(2) ≤ 3, NSSPTS ≤ 3
58	PAWS (D,3,0)	ERROR IN INPUT PARAMETER ARRAY TO IMSL SUBROUTINE FTFREQ. IND(4) ≤ 2, M ≤ 2
59	PAWS (D,3,0)	ERROR IN INPUT PARAMETER ARRAY TO IMSL SUBROUTINE FTFREQ. IND(4) ≥ IND(2), M ≥ NSSPTS

EXECUTION ERROR CODE DESCRIPTION (cont.)

PERROR ERROR CODE	SUBROUTINE/OVERLAY	ERROR DESCRIPTION
60	SUBSET (C,2,0)	CALCULATION ERROR IN DATA SUBSET INITIAL POINT. CALCULATED POINT DOES NOT CORRESPOND TO ACTUAL INITIAL POINT
61	SUBSET (C,2,0)	CALCULATION ERROR IN DATA SUBSET FINAL POINT. CALCULATED POINT DOES NOT CORRESPOND TO ACTUAL FINAL POINT.
63	THEFFT (F,5,0)	UNABLE TO REPLACE DATA INTO DATA RECORD. SDTIME $\neq$ DTIME
64	THEFFT (F,5,0)	CALCULATION ERROR IN DATA SUBSET INITIAL POINT. CALCULATED POINT DOES NOT CORRESPOND WITH ACTUAL INITIAL POINT.
71	COMPLT (A,0,0)	IMPROPER REGIME I DIVISION OPTION. IDIV $\neq$ -1, 0, 1
73	COMPLT (A,0,0)	IMPROPER DIVISION TIME OF REGIME I. SUBTIM NOT IN REGIME I.
79	KITTY8 (A,0,0)	IMPROPER OPTION FOR ANALYSIS OF DATA RECORD OR DATA SUBSET. SUBOPT < 0 SUBOPT $\neq$ 1, 2, 3, 4, 5, 11, OR 12
81	THEFFT (F,5,0)	ERROR IN ARRAY DIMENSION DATA SUBSET. ARRAY DIMENSION TOO LARGE
82	SCRIB1 (K,3,1)	ERROR IN ARRAY DIMENSION. NDPTS > 100, TOO MANY POINTS TO PLOT
83	SCRIBL (J,5,3)	ERROR IN INPUT TO PLOTTING ROUTINE. NARRAY < 1 OR NARRAY > 3, NUMBER OF DATA ARRAYS TO PLOT ON A GRAPH

EXECUTION ERROR CODE DESCRIPTION (cont.)

PERROR ERROR CODE	SUBROUTINE/OVERLAY	ERROR DESCRIPTION
84	SCRIBL (J,5,3)	ERROR IN ARRAY DIMENSION. NDPTS > 5000, TOO MANY POINTS TO PLOT
85	THEFFT (F,5,0)	EXECUTION ERROR IN IMSL SUBROUTINE GIFT. IER > 0, SEE IMSL MANUAL.
87	POLYOR (H,5,1)	EXECUTION ERROR IN IMSL SUBROUTINE RLFOR. IER = 33, SEE IMSL MANUAL
88	POLYOR (H,5,1)	EXECUTION ERROR IN IMSL SUBROUTINE RLFOR. IER > 0, SEE IMSL MANUAL
90	PAWS (I,5,2)	ERROR IN INPUT PARAMETER ARRAY TO IMSL SUBROUTINE FTFREQ. IND(2) $\leq$ 3, NSSPTS $\leq$ 3
91	PAWS (I,5,2)	ERROR IN INPUT PARAMETER ARRAY TO IMSL SUBROUTINE FTFREQ. IND(4) $\leq$ 2, M $\leq$ 2
92	PAWS (I,5,2)	ERROR IN INPUT PARAMETER ARRAY TO IMSL SUBROUTINE FTFREQ. IND(4) $\geq$ IND(2), M $\geq$ NSSPTS
93	SCRIB2 (I,5,2)	ERROR IN ARRAY DIMENSION. NDPTS > 100, TOO MANY POINTS TO PLOT
94	SMOOTH (G,6,0)	IMPROPER SMOOTHING OPTION. SMOPT < 1, OR SMOPT > 2
95	SCRIB4 (M,7,1)	ERROR IN ARRAY DIMENSION. NDPTS > 5000, TOO MANY POINTS TO PLOT

DISTRIBUTION LIST

No. of Copies	Organization
12	Administrator Defense Technical Info Center ATTN: DTIC-DDA Cameron Station Alexandria, VA 22314
2	Director Defense Advanced Research Projects Agency 1400 Wilson Boulevard Arlington, VA 22209
1	Director Defense Nuclear Agency Washington, DC 20305
1	Commander US Army BMD Advanced Technology Center ATTN: BMDATC-M P.O. Box 1500 Huntsville, AL 35804
1	Commander US Army Materiel Development and Readiness Command ATTN: DRCDMD-ST 5001 Eisenhower Avenue Alexandria, VA 22333
6	Commander USA ARRADCOM ATTN: DRDAR-TSS (2 cys) DRDAR-TD DRDAR-TDA DRDAR-TDS DRDAR-TDC Dover, NJ 07801
9	Commander USA ARRADCOM ATTN: DRDAR-LCA DRDAR-LCW DRDAR-LCE DRDAR-LCM DRDAR-LCN DRDAR-LCU DRDAR-LC DRDAR-LCS-D, Mr. Kenneth Rubin MAJ John L. Houle Dover, NJ 07801
6	Commander USA ARRADCOM ATTN: DRDAR-SC DRDAR-SA DRDAR-SCM DRDAR-SCS DRDAR-SCA DRDAR-SCF Dover, NJ 07801
2	Commander USA ARRADCOM ATTN: DRDAR-SE Dover, NJ 07801
2	Commander USA ARRADCOM Product Assurance Directorate ATTN: DRDAR-QA Dover, NJ 07801
1	Commander USA ARRADCOM ATTN: Army Fuze Mgt Project Office DRDAR-FU Dover, NJ 07801
2	Commander USA ARRADCOM ATTN: Development Project Office for Selected Ammunition DRDAR-DP Dover, NJ 07801
1	Director Benet Weapons Laboratory USA ARRADCOM ATTN: DRDAR-LCB-TL Watervliet, NY 12189
1	Commander US Army Rock Island Arsenal ATTN: DRDAR-TSE-SW, R. Radkiewicz Rock Island, IL 61299
1	Commander US Army Armament Materiel Readiness Command ATTN: DRSAR-LEP-L, Tech Lib Rock Island, IL 61299

DISTRIBUTION LIST

No. of Copies	Organization
2	Commander US Army Aviation Research and Development Command ATTN: DRDAV-E 4300 Goodfellow Blvd. St. Louis, MO 63120
1	Director US Army Air Mobility Research and Development Laboratory Ames Research Center Moffett Field, CA 94035
2	Director US Army Research and Technology Laboratories (AVRADCOM) Ames Research Center Moffett Field, CA 94035
1	Commander US Army Communications Rsch and Development Command ATTN: DRDCO-PPA-SA Fort Monmouth, NJ 07703
1	Commander US Army Electronics Research and Development Command Technical Support Activity ATTN: DELSD-L Fort Monmouth, NJ 07703
3	Commander US Army Harry Diamond Labs ATTN: H. Davis H. Curchack DELHD-S-OE-ES, Ben Banner 2800 Powder Mill Road Adelphi, MD 20783
1	Commander US Army Missile Command ATTN: DRSMI-R Redstone Arsenal, AL 35898
1	Commander US Army Missile Command ATTN: DRSMI-YDL Redstone Arsenal, AL 35898
1	Commander US Army Missile Command ATTN: DRSMI-RBL Redstone Arsenal, AL 35898
2	Commander US Army Tank Automotive Research and Development Command ATTN: DRDTA-UL Technical Director Warren, MI 48090
2	Commander US Army Jefferson Proving Ground ATTN: STEJP-TD-0, Arnold Tilley STEJP-TD-E, Joseph Tooney Madison, IN 47251
2	Commander US Army Yuma Proving Ground ATTN: STEYP-MTW, Robert Torp Graham Stullenbarger Yuma, AZ 85364
2	President US Army Armor and Engineer Board ATTN: ATZK-RE-CV ATZK-RE-IN, Mr. Larry Smith Fort Knox, KY 40121
2	Project Manager Division Air Defense Gun ATTN: DRCPM-ADG Dover, NJ 07801
3	Project Manager Cannon Artillery Weapons System ATTN: DRCPM-CAWS Dover, NJ 07801
2	Project Manager Nuclear Munitions ATTN: DRCPM-NUC Dover, NJ 07801
2	Project Manager Tank Main Armament Systems ATTN: DRCPM-TMA Dover, NJ 07801

DISTRIBUTION LIST

No. of Copies	Organization
1	Product Manager for 30mm Ammo. ATTN: DRCPM-AAH-30mm Dover, NJ 07801
2	Product Manager M110E2 Weapon System, DARCOM ATTN: DRCPM-M110E2 Rock Island, IL 61299
4	Director USA Mechanics and Materials Research Center ATTN: Director (3 cy) DRXMR-ATL (1 cy) Watertown, MA 02172
3	Commander US Army Research Office P.O. Box 12211 ATTN: Technical Director Engineering Division Metallurgy & Materials Division Research Triangle Park NC 27709
1	Director US Army TRADOC Systems Analysis Activity ATTN: ATAA-SL (Tech Lib) White Sands Missile Range, NM 88002
1	Commander Naval Air Systems Command Washington, DC 20360
1	Commander Naval Sea Systems Command Washington, DC 20362
1	Commander Naval Sea Systems Command ATTN: SEA-03513, L. Pasiuk Washington, DC 20362
1	Commander Naval Air Development Center Johnsville Warminster, PA 18974

No. of Copies	Organization
1	Commander Naval Missile Center Point Mugu, CA 93041
2	Commander David W. Taylor Naval Ship Research & Development Center Bethesda, MD 20084
2	Commander Naval Surface Weapons Center Dahlgren, VA 22448
4	Commander Naval Surface Weapons Center ATTN: Code G-33, T.N. Tschirn Code N-43, J.J. Yagla L. Anderson G. Soo Hoo Dahlgren, VA 22448
3	Commander Naval Surface Weapons Center ATTN: Code E-31, R.C. Reed M.T. Walchak Code V-14, W.M. Hinckley Silver Spring, MD 20910
2	Commander Naval Surface Weapons Center Silver Spring, MD 20910
2	Commander Naval Weapons Center China Lake, CA 93555
3	Commander Naval Weapons Center ATTN: J. O'Malley D. Potts R.G. Sewell China Lake, CA 93555
1	Commander Naval Research Laboratory Washington, DC 20375

DISTRIBUTION LIST

No. of Copies	Organization
1	Superintendent Naval Postgraduate School ATTN: Dir of Lib Monterey, CA 93940
1	Commander Naval Ordnance Station Indian Head, MD 20640
2	Commander Naval Ordnance Station ATTN: Code 5034, Ch, Irish Jr. T.C. Smith Indian Head, MD 20640
2	AFATL Eglin AFB, FL 32542
3	AFATL (DLA, W. Dittrich; DLD, D. Davis; DLDL) Eglin AFB, FL 32542
2	AFWL/SUL Kirtland AFB, NM 87115
2	ASD Wright-Patterson AFB, OH 45433
1	Director Lawrence Livermore Laboratory P. O. Box 808 Livermore, CA 94550
1	Director Los Alamos Scientific Lab P. O. Box 1663 Los Alamos, NM 87544
2	Director National Aeronautics and Space Administration Langley Research Center Langley Station Hampton, VA 23365
1	Aircraft Armaments Inc. ATTN: John Hebert York Road 7 Industry Lane Cockeysville, MD 21030

No. of Copies	Organization
2	ARES Inc. ATTN: Duane Summers Phil Conners Port Clinton, OH 43452
1	General Electric Company ATTN: Armament Systems Dept. David A. Graham Lakeside Avenue Burlington, VT 05402
1	Falcon Research and Development Co. 696 Fairmount Avenue Baltimore, MD 21204

Aberdeen Proving Ground

Dir, USAMSAA
ATTN: DRXSY-D
DRXSY-G, E. Christman
DRXSY-OSD, H. Burke
DRXSY-G, R. C. Conroy
DRXSY-LM, J. C. C. Fine
DRXSY-MP, H. Cohen

Dir, USAHEL
ATTN: A. H. Eckles, III
Cdr, USATECOM
ATTN: DRSTE-TO-F
Dir, USACSL, Bldg. E3516, EA
ATTN: DRDAR-CLB-PA
DRDAR-CL
DRDAR-CLB
DRDAR-CLD
DRDAR-CLY
DRDAR-CLN

USER EVALUATION OF REPORT

Please take a few minutes to answer the questions below; tear out this sheet, fold as indicated, staple or tape closed, and place in the mail. Your comments will provide us with information for improving future reports.

1. BRL Report Number ______________________________

2. Does this report satisfy a need? (Comment on purpose, related project, or other area of interest for which report will be used.)

____

____

____

3. How, specifically, is the report being used? (Information source, design data or procedure, management procedure, source of ideas, etc.) ______________________________

____

____

4. Has the information in this report led to any quantitative savings as far as man-hours/contract dollars saved, operating costs avoided, efficiencies achieved, etc.? If so, please elaborate.

____

____

5. General Comments (Indicate what you think should be changed to make this report and future reports of this type more responsive to your needs, more usable, improve readability, etc.) ____________

____

____

____

6. If you would like to be contacted by the personnel who prepared this report to raise specific questions or discuss the topic, please fill in the following information.

Name: ______________________________

Telephone Number: ______________________________

Organization Address: ______________________________

____

____

— — — — — — — — — FOLD HERE — — — — — — — — — — —

Director
US Army Ballistic Research Laboratory
Aberdeen Proving Ground, MD 21005

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

OFFICIAL BUSINESS

PENALTY FOR PRIVATE USE, $300

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO 12062 WASHINGTON, DC

POSTAGE WILL BE PAID BY DEPARTMENT OF THE ARMY

Director
US Army Ballistic Research Laboratory
ATTN: DRDAR-TSB
Aberdeen Proving Ground, MD 21005

— — — — — — — — FOLD HERE — — — — — — — — — — —